$99

TRSDOS/LS - DOS 6.x

"THE SOURCE"

```
LD      A,(BUFFER$+1)    ;P/u buffer hi-order addr
LD      D,A
LD      BC,13            ;Move name/ext into dest
LDIR
LD      D,(IY+9)         ;P/u dir cyl of dest
POP     BC               ;Rcvr DEC of source
PUSH    BC
LD      A,B              ;Calc dir sector for
AND     1FH              ;  source SYS module
ADD     A,2
LD      E,A
LD      HL,(BUFFER$)     ;P/u buffer ptr for dest
CALL    WRSYS            ;Write the dir to dest
LD      A,18             ;Init "Dir write error
JP      NZ,EXIT3         ;  and quit on bad write

The HIT entries were transferred prior

POP     BC               ;Rcvr DEC of source
PUSH    BC
LD      A,B              ;Test for SYSØ
CP      2
JP      NZ,DOFILØ        ;Bypass if not SYSØ
CALL    PMTSRC           ;Prompt source
IF      @MOD4
LD      B,16             ;Init to xfer BOOT track
LD      DE,Ø             ;Init track Ø, sector Ø
ENDIF
IF      @MOD2
LD      DE,(PROTSEC)     ;Get sysinfo sector
LD      A,D
OR      A
LD      B,5
```

LOGICAL SYSTEMS INC.

Volume 3

The UTILITIES

CONTENTS

LOGICAL SYSTEMS INC.

8970 North 55th Street
P.O. Box 23956
Milwaukee, WI 53223

Introduction to Volume Three

This is volume three of three in the set of commented source code listings for LS-DOS/TRSDOS 6.2, as assembled for the TRS-8Ø Model 4/4P computer. This volume contains the utility programs, the standard drivers and filters, and the drive setup programs.

Each file will be preceded by a brief description of its function. A symbol table listing will follow each assembly listing.

The SVC macro file used during assembly of the utilities will be listed in Appendix A, along with the equate files generated during assembly of the resident part of the operating system.

This book should by no means be considered a tutorial on assembly language or on the workings of the LS-DOS/TRSDOS operating system. It is only the commented source code used to assemble the system utilities. It can be used for reference purposes and to view examples of interfacing outside drivers, filters and other programs to the DOS and to each other. It is not meant to replace the normal technical reference manual available from the computer manufacturer.

This product is sold on an as-is basis, and is totally unsupported by Logical Systems, Inc. No questions regarding any aspect of the source code will be answered by LSI customer or technical support. Support for LS-DOS users is provided through their OEM dealer. Support for TRSDOS 6.x is provided by Tandy Corporation. Comments or suggestions may be sent to Logical Systems, Inc. in care of the Source Code Technical Editor, but correspondence concerning these comments will not be made.

BACKUP/CMD - Disk and file duplication utility

Backup is assembled from three separate source files. The first contains some initialization, and the common code needed during mirror image backups and backup by class. The other modules contain the code for the two different type of backup operations.

```
                ;BACKUP/ASM - File/disk copy utility
0000                    TITLE   <BACKUP - LS-DOS 6.2>
                ;
0000            *GET    BACKUP1:3
                ;BACKUP1/ASM - Backup utility module
0000                    SUBTTL  '<Backup initialization>'
                ;
0000            SMALL   EQU     0
0000            FMT     EQU     0
000A            LF      EQU     10
000D            CR      EQU     13
0060            LOCK    EQU     60H
00CC            TKCAP   EQU     0CCH
00CE            PSWD    EQU     0CEH
00D8            DAT     EQU     0D8H
00E0            AUTO    EQU     0E0H
1111            FCNT1   EQU     1111H
1555            FCNT2   EQU     1555H
42E0            PASSWORD        EQU     42E0H
                ;
0000            *GET    SVCMAC:3                ;SVC Macro equivalents
                ;SVCMAC/ASM - LS-DOS Version VI
                *LIST   OFF
                *LIST   ON
0000            *GET    COPYCOM:3               ;Copyright message
                ; COPYCOM - File for Copyright COMment block
                ;
0000                    COM     '<*(C) 1982,83,84 by LSI*>'
                ;
2600                    ORG     2600H
                ;
                        IF      @MOD2
                BOOTST$ DB      03H
                        ENDIF
                        IF      @MOD4
2600 9D         BOOTST$ DB      9DH             ;Boot step rate ptr
                        ENDIF
                ;
                ;       Data area
                ;
2601 00         FTFLG$  DB      0
2602 20         SPCFLD$ DC      11,' '
     20 20 20 20 20 20 20 20
     20 20
260D 00         MFLG$   DB      0
260E 0000       NEWPRM$ DW      0
2610 0000       OLDPRM$ DW      0
2612 0000       MODPRM$ DW      0
2614 0000       QPARM$  DW      0
2616 0000       BUFFER$ DW      0
0020            FCB1$   DS      32
0020            FCB2$   DS      32
0020            FCB3$   DS      32
2658            LILBUF$ EQU     FCB3$
0008            DATFLD$ DS      8
0002            FMPAKD$ DS      2
0002            TOPAKD$ DS      2
0001            CLSFLG$ DS      1
                ;
                        IF      @MOD2
                ;
```

Backup initialization

```
                ØØ444Ø          SUBTTL  '<BACKUP Module - #4/2>'
                Ø445Ø ;
                Ø447Ø           ENDIF
                Ø448Ø ;
                Ø449Ø ;
                Ø45ØØ ;         Normal exit - no errors
                Ø451Ø ;
2685 21FC29     Ø452Ø EXIT1     LD      HL,BUCAO$          ;"Backup complete...
2688 E5         Ø453Ø           PUSH    HL                 ;Save msg ptr
2689 CDC926     Ø454Ø           CALL    EXIT5              ;Ck if prompt for sys disk
268C E1         Ø455Ø           POP     HL
268D            Ø456Ø           @@DSPLY
                ØØØØ1           IFEQ    ØØH,1
                ØØØØ2           LD      HL,
                ØØØØ3           ENDIF
268D 3EØA       ØØØØ4           LD      A,1Ø
268F EF         ØØØØ5           RST     4Ø
269Ø 182B       Ø457Ø           JR      EXIT
                Ø458Ø ;
                Ø459Ø ;         Error exit
                Ø46ØØ ;
2692 3E11       Ø461Ø DIRERR    LD      A,17               ;Init "Dir read error
2694 Ø1         Ø462Ø           DB      1                  ;Ignore next inst
2695 3E2Ø       Ø463Ø EXIT2     LD      A,2ØH              ;Init illegal drive #
2697 F5         Ø464Ø EXIT3     PUSH    AF
2698 ØEØD       Ø465Ø           LD      C,CR               ;Terminate pending line
269A            Ø466Ø           @@DSP
269A 3EØ2       ØØØØ6           LD      A,2
269C EF         ØØØØ7           RST     4Ø
269D CDC926     Ø467Ø           CALL    EXIT5              ;Get system disk if needed
26AØ F1         Ø468Ø           POP     AF
26A1 6F         Ø469Ø           LD      L,A                ;Error code to HL
26A2 26ØØ       Ø47ØØ           LD      H,Ø
26A4 F6CØ       Ø471Ø           OR      ØCØH               ;Set short,return
26A6 4F         Ø472Ø           LD      C,A                ;Error to C
26A7            Ø473Ø           @@ERROR                    ;  for error dsply
26A7 3E1A       ØØØØ8           LD      A,26
26A9 EF         ØØØØ9           RST     4Ø
26AA 18ØE       Ø474Ø           JR      ERREXIT
                Ø475Ø ;
                Ø476Ø ;         Abort exit
                Ø477Ø ;
26AC            Ø478Ø BREAK     EQU     $
26AC 21ØD2A     Ø479Ø ABRTBU    LD      HL,ABRTBU$         ;"Backup aborted
26AF E5         Ø48ØØ EXIT4     PUSH    HL                 ;Save msg ptr
26BØ CDC926     Ø481Ø           CALL    EXIT5              ;Get system disk if needed
26B3 E1         Ø482Ø           POP     HL
26B4            Ø483Ø           @@LOGOT                    ;Display the message
                ØØØ1Ø           IFEQ    ØØH,1
                ØØØ11           LD      HL,
                ØØØ12           ENDIF
26B4 3EØC       ØØØ13           LD      A,12
26B6 EF         ØØØ14           RST     4Ø
26B7 21FFFF     Ø484Ø           LD      HL,-1              ;Set error return code
26BA 22C126     Ø485Ø ERREXIT   LD      (RETCOD),HL
26BD            Ø486Ø EXIT      EQU     $
26BD 31ØØØØ     Ø487Ø SPSAV     LD      SP,$-$             ;P/u the stack pointer
26CØ 21ØØØØ     Ø488Ø           LD      HL,Ø               ;Set the return code
26C1            Ø489Ø RETCOD    EQU     $-2
```

Backup initialization

```
26C3           Ø49ØØ          @@CKBRKC                ;Check and clear break
26C3 3E6A      ØØØ15          LD      A,1Ø6
26C5 EF        ØØØ16          RST     4Ø
26C6           Ø491Ø          @@EXIT                  ;Can't return from BACKUP
26C6 3E16      ØØØ17          LD      A,22
26C8 EF        ØØØ18          RST     4Ø
               Ø492Ø ;
               Ø493Ø ;        Get system disk if needed & zero memory used
               Ø494Ø ;
26C9           Ø495Ø EXIT5    EQU     $
26C9 11ØØØØ    Ø496Ø XPARM$   LD      DE,Ø            ;P/u prompt zero drive
26CC 1C        Ø497Ø          INC     E               ;Ck for entry
26CD 2ØØ9      Ø498Ø          JR      NZ,EXIT5A
26CF AF        Ø499Ø          XOR     A
26DØ 32CF27    Ø5ØØØ          LD      (SXORD+1),A
26D3 CDØØ27    Ø5Ø1Ø          CALL    SYSDRV$         ;Force prompt for SYSTEM
26D6 18Ø7      Ø5Ø2Ø          JR      EXIT5B
26D8 3ACF27    Ø5Ø3Ø EXIT5A   LD      A,(SXORD+1)     ;  else if not entered,
26DB B7        Ø5Ø4Ø          OR      A               ;  ck if source & dest
26DC CCFB26    Ø5Ø5Ø          CALL    Z,NDSYS$        ;  are same - we may need
26DF ED5B1626  Ø5Ø6Ø EXIT5B   LD      DE,(BUFFER$)    ;  a prompt
26E3 7A        Ø5Ø7Ø          LD      A,D             ;Ck if we did a backup
26E4 B3        Ø5Ø8Ø          OR      E
26E5 C8        Ø5Ø9Ø          RET     Z               ;Ret if buf adr never set
26E6 21ØØØØ    Ø51ØØ          LD      HL,Ø            ;  else calculate how
26E9 45        Ø511Ø          LD      B,L             ;  many bytes in RAM
26EA           Ø512Ø          @@HIGH$                 ;  to zero
26EA 3E64      ØØØ19          LD      A,1ØØ
26EC EF        ØØØ2Ø          RST     4Ø
26ED AF        Ø513Ø          XOR     A
26EE ED52      Ø514Ø          SBC     HL,DE           ;Get length to zero
26FØ 44        Ø515Ø          LD      B,H
26F1 4D        Ø516Ø          LD      C,L             ;  into BC
26F2 62        Ø517Ø          LD      H,D             ;Pt HL to start of buffer
26F3 6B        Ø518Ø          LD      L,E
26F4 13        Ø519Ø          INC     DE
26F5 36ØØ      Ø52ØØ          LD      (HL),Ø          ;Init 1st to zero
26F7 ØB        Ø521Ø          DEC     BC              ;  & propogate it
26F8 EDBØ      Ø522Ø          LDIR
26FA C9        Ø523Ø          RET
               Ø524Ø ;
               Ø525Ø ;        Prompt for system disk
               Ø526Ø ;
26FB 3AØE27    Ø527Ø NDSYS$   LD      A,(SRCDRV$+1)   ;On exit, if S=D <> Ø
26FE B7        Ø528Ø          OR      A               ;  then no need to prompt
26FF CØ        Ø529Ø          RET     NZ
27ØØ 3EØØ      Ø53ØØ SYSDRV$  LD      A,Ø             ;P/u drive Ø indicator
27Ø2 F62Ø      Ø531Ø          OR      2ØH             ;Set bit 5 for sys test
27Ø4 E5        Ø532Ø          PUSH    HL
27Ø5 21FE28    Ø533Ø          LD      HL,PMTSYS$      ;"insert system...
27Ø8 CDC727    Ø534Ø          CALL    CURDSK
27ØB E1        Ø535Ø          POP     HL
27ØC C9        Ø536Ø          RET
               Ø537Ø ;
               Ø538Ø ;        Prompt source disk
               Ø539Ø ;
27ØD 3EØØ      Ø54ØØ SRCDRV$  LD      A,Ø             ;Source drive
27ØF F68Ø      Ø541Ø          OR      8ØH             ;Set bit 7 on source
2711 E5        Ø542Ø          PUSH    HL
```

Backup initialization

```
2712 211C29                LD      HL,PMTSRC$        ;"Insert source
2715 CDC727                CALL    CURDSK            ;Prompt for source if needed
2718 E1                    POP     HL
2719 C9                    RET
                   ;
                   ;       Prompt source disk if needed to swap
                   ;
271A 3AC827        PMTSRC  LD      A,(CURDSK+1)      ;P/u current drive
271D CB7F                  BIT     7,A               ;Is source the one?
271F 2ØEC                  JR      NZ,SRCDRV$        ;Jump if it is
2721 CDØD27                CALL    SRCDRV$           ;  else prompt for it
2724 3ACF27                LD      A,(SXORD+1)
2727 B7                    OR      A
2728 CØ                    RET     NZ                ;Ret if source <> dest
2729 CD5E28                CALL    RESTOR            ;Restore to cyl Ø
272C C5                    PUSH    BC
272D D5                    PUSH    DE                ;Save registers
272E E5                    PUSH    HL
272F 21ØØ2D                LD      HL,BUF3$          ;Use this for I/O buffer
2732 11ØØØØ                LD      DE,Ø              ;Read the BOOT
2735 CD7228                CALL    RDSEC
2738 E1                    POP     HL
2739 D1                    POP     DE                ;Restore the registers
273A C1                    POP     BC
273B C29726                JP      NZ,EXIT3          ;Quit on read error
273E 3AØØ2D                LD      A,(BUF3$)         ;P/u 1st byte of BOOT
2741 B7                    OR      A                 ;If source, s/b Ø
2742 2Ø3Ø                  JR      NZ,PSRC3          ;Jump if not this disk
2744 C5                    PUSH    BC
2745 D5                    PUSH    DE
2746 E5                    PUSH    HL
2747 FD56Ø9                LD      D,(IY+9)          ;P/u dir cyl
274A 1EØØ                  LD      E,Ø               ;Pt to GAT sector
274C 21ØØ2D                LD      HL,BUF3$
274F CD7228                CALL    RDSEC             ;Read the GAT
2752 FEØ6                  CP      6
2754 C29226                JP      NZ,DIRERR
2757 21CE2B                LD      HL,BUF1$+PSWD     ;Ck for match with orig
275A 11CE2D                LD      DE,BUF3$+PSWD     ;  source disk
275D Ø6ØA                  LD      B,1Ø              ;Set match count
275F 1A            PSRC1   LD      A,(DE)
276Ø BE                    CP      (HL)
2761 2ØØ8                  JR      NZ,DIFSRC         ;Wrong disk if no match
2763 13                    INC     DE                ;Bump pointers
2764 23                    INC     HL
2765 1ØF8                  DJNZ    PSRC1             ;Loop for 1Ø compares
2767 E1                    POP     HL                ;Was a match,
2768 D1                    POP     DE                ;  restore and return
2769 C1                    POP     BC
276A C9                    RET
276B               DIFSRC  @@DSPLY DIFSRC$           ;"wake up...
                           IFEQ    Ø1H,1
276B 215D29                LD      HL,DIFSRC$
                           ENDIF
276E 3EØA                  LD      A,1Ø
277Ø EF                    RST     4Ø
2771 E1                    POP     HL                ;Clean the stack
2772 D1                    POP     DE
2773 C1                    POP     BC
```

Backup initialization

```
2774 AF         Ø597Ø PSRC3   XOR     A               ;Show not current disk
2775 32C827     Ø598Ø         LD      (CURDSK+1),A
2778 18AØ       Ø599Ø         JR      PMTSRC          ;Loop to re-prompt
                Ø6ØØØ ;
                Ø6Ø1Ø ;       Destination disk selection
                Ø6Ø2Ø ;
277A 3EØØ       Ø6Ø3Ø DSTDRV$ LD      A,Ø             ;Dest drive
277C F64Ø       Ø6Ø4Ø         OR      4ØH             ;Set dest diskette code
277E E5         Ø6Ø5Ø         PUSH    HL
277F 213A29     Ø6Ø6Ø         LD      HL,PMTDST$      ;"insert dest...
2782 CDC727     Ø6Ø7Ø         CALL    CURDSK
2785 E1         Ø6Ø8Ø         POP     HL
2786 C9         Ø6Ø9Ø         RET
                Ø61ØØ ;
                Ø611Ø ;       Prompt destination if needed
                Ø612Ø ;
2787 3AC827     Ø613Ø PMTDST  LD      A,(CURDSK+1)    ;P/u current disk/drive &
278A CB77       Ø614Ø         BIT     6,A             ;  ck if destination disk
278C 2ØEC       Ø615Ø         JR      NZ,DSTDRV$      ;Jump if it is
278E CD7A27     Ø616Ø         CALL    DSTDRV$         ;  else request swap
2791 3ACF27     Ø617Ø         LD      A,(SXORD+1)
2794 B7         Ø618Ø         OR      A
2795 CØ         Ø619Ø         RET     NZ              ;Ret if source <> dest
2796 CD5E28     Ø62ØØ         CALL    RESTOR          ;  else restore to cyl Ø
2799 C5         Ø621Ø         PUSH    BC
279A D5         Ø622Ø         PUSH    DE
279B E5         Ø623Ø         PUSH    HL
279C 21ØØ2D     Ø624Ø         LD      HL,BUF3$        ;Use this for I/O buffer
279F 11ØØØØ     Ø625Ø         LD      DE,Ø            ;Pt to BOOT sector
27A2 CD7228     Ø626Ø         CALL    RDSEC           ;  & read the BOOT
27A5 E1         Ø627Ø         POP     HL
27A6 D1         Ø628Ø         POP     DE
27A7 C1         Ø629Ø         POP     BC
27A8 C29726     Ø63ØØ         JP      NZ,EXIT3        ;Quit on read error
27AB 3AØØ2D     Ø631Ø         LD      A,(BUF3$)       ;P/u 1st byte of BOOT
27AE FE76       Ø632Ø         CP      76H             ;Dest s/b a HALT
27BØ C8         Ø633Ø PMTDST1 RET     Z
27B1 E5         Ø634Ø         PUSH    HL
27B2 D5         Ø635Ø         PUSH    DE
27B3            Ø636Ø         @@DSPLY DIFDST$         ;"not same dest...
                ØØØ26         IFEQ    Ø1H,1
27B3 218F29     ØØØ27         LD      HL,DIFDST$
                ØØØ28         ENDIF
27B6 3EØA       ØØØ29         LD      A,1Ø
27B8 EF         ØØØ3Ø         RST     4Ø
27B9 D1         Ø637Ø         POP     DE
27BA E1         Ø638Ø         POP     HL
27BB AF         Ø639Ø         XOR     A
27BC 32C827     Ø64ØØ         LD      (CURDSK+1),A    ;Show no current diskette
27BF 18C6       Ø641Ø         JR      PMTDST          ;  and prompt again
                Ø642Ø ;
                Ø643Ø ;       Force a prompt of the target disk
                Ø644Ø ;
27C1 79         Ø645Ø FRCPMT  LD      A,C             ;P/u target drive
27C2 32C827     Ø646Ø         LD      (CURDSK+1),A    ; with code bit
27C5 18ØC       Ø647Ø         JR      FLASH
                Ø648Ø ;
                Ø649Ø ;       Routine to check if flashing prompt is needed
                Ø65ØØ ;
```

Backup initialization

```
27C7 FEØØ        Ø651Ø CURDSK   CP       Ø                 ;P/u current disk
27C9 2872        Ø652Ø          JR       Z,FLSH6           ;Match with wanted disk?
27CB 32C827      Ø653Ø          LD       (CURDSK+1),A      ;No, update current
27CE 3EFF        Ø654Ø SXORD    LD       A,ØFFH            ;Ø=src & dst drive same
27DØ B7          Ø655Ø          OR       A
27D1 2Ø6A        Ø656Ø          JR       NZ,FLSH6          ;Jump if source <> dest
                 Ø657Ø ;
                 Ø658Ø ;        Routine to flash the prompt
                 Ø659Ø ;
27D3 C5          Ø66ØØ FLASH    PUSH     BC
27D4 D5          Ø661Ø          PUSH     DE
27D5 E5          Ø662Ø          PUSH     HL
27D6             Ø663Ø          @@FLAGS                    ;IY => flag table base
27D6 3E65        ØØØ31          LD       A,1Ø1
27D8 EF          ØØØ32          RST      4Ø
27D9 ØEØD        Ø664Ø          LD       C,CR              ;Write a new line
27DB             Ø665Ø          @@DSP
27DB 3EØ2        ØØØ33          LD       A,2
27DD EF          ØØØ34          RST      4Ø
27DE ØEØF        Ø666Ø          LD       C,15              ;Cursor off
27EØ             Ø667Ø          @@DSP
27EØ 3EØ2        ØØØ35          LD       A,2
27E2 EF          ØØØ36          RST      4Ø
                 Ø668Ø FLASHØ
27E3             Ø669Ø          @@CKBRKC                   ;Check and clear break
27E3 3E6A        ØØØ37          LD       A,1Ø6
27E5 EF          ØØØ38          RST      4Ø
27E6 CD4C28      Ø67ØØ          CALL     RESKFLG           ;Reset Pause,Enter,Break
27E9 Ø1FD41      Ø671Ø          LD       BC,16893          ;Delay for 1/4 sec
27EC             Ø672Ø          @@PAUSE
27EC 3E1Ø        ØØØ39          LD       A,16
27EE EF          ØØØ4Ø          RST      4Ø
27EF FD7EØA      Ø673Ø          LD       A,(IY+'K'-'A')
27F2 E6Ø5        Ø674Ø          AND      4!1               ;Wait for no ENTER!BRK
27F4 2ØED        Ø675Ø          JR       NZ,FLASHØ
27F6 CD4C28      Ø676Ø          CALL     RESKFLG           ;Reset in case BREAK
27F9             Ø677Ø FLS1     @@DSPLY                    ;Display the message
                 ØØØ41          IFEQ     ØØH,1
                 ØØØ42          LD       HL,
                 ØØØ43          ENDIF
27F9 3EØA        ØØØ44          LD       A,1Ø
27FB EF          ØØØ45          RST      4Ø
27FC Ø15515      Ø678Ø          LD       BC,FCNT2
27FF CD1428      Ø679Ø          CALL     FLS2              ;Blink start
28Ø2 ØE1D        Ø68ØØ          LD       C,29              ;Cursor to BOL
28Ø4             Ø681Ø          @@DSP
28Ø4 3EØ2        ØØØ46          LD       A,2
28Ø6 EF          ØØØ47          RST      4Ø
28Ø7 ØE1E        Ø682Ø          LD       C,1EH             ;Cursor erase to EOL
28Ø9             Ø683Ø          @@DSP
28Ø9 3EØ2        ØØØ48          LD       A,2
28ØB EF          ØØØ49          RST      4Ø
28ØC Ø11111      Ø684Ø          LD       BC,FCNT1          ;Wait delay count
28ØF CD1428      Ø685Ø          CALL     FLS2              ;Wait & ck Enter or Break
2812 18E5        Ø686Ø          JR       FLS1              ;Loop until Enter
                 Ø687Ø FLS2
2814             Ø688Ø          @@CKBRKC                   ;Check for break
2814 3E6A        ØØØ5Ø          LD       A,1Ø6
2816 EF          ØØØ51          RST      4Ø
```

Backup initialization

```
2817 C2AC26     Ø689Ø          JP      NZ,BREAK          ;  and abort if so
281A FD7EØA     Ø69ØØ          LD      A,(IY+'K'-'A')    ;P/u KFLAG settings
281D CB57       Ø691Ø          BIT     2,A               ;Enter pressed?
281F 2ØØ6       Ø692Ø          JR      NZ,FLS4           ;Go if so
2821 ØB         Ø693Ø          DEC     BC                ;Count down
2822 78         Ø694Ø          LD      A,B
2823 B1         Ø695Ø          OR      C
2824 2ØEE       Ø696Ø          JR      NZ,FLS2           ;  and loop if more time
2826 C9         Ø697Ø          RET
2827 F1         Ø698Ø FLS4     POP     AF                ;Pop return address
2828            Ø699Ø FLS5     @@KBD                     ;Clear type ahead buffer
2828 3EØ8       ØØØ52          LD      A,8
282A EF         ØØØ53          RST     4Ø
282B 28FB       Ø7ØØØ          JR      Z,FLS5            ;Loop til no key down
282D ØEØD       Ø7Ø1Ø          LD      C,ØDH             ;Dsply a new line
282F            Ø7Ø2Ø          @@DSP
282F 3EØ2       ØØØ54          LD      A,2
2831 EF         ØØØ55          RST     4Ø
2832 ØEØE       Ø7Ø3Ø          LD      C,14              ;Cursor on
2834            Ø7Ø4Ø          @@DSP
2834 3EØ2       ØØØ56          LD      A,2
2836 EF         ØØØ57          RST     4Ø
2837 CD4C28     Ø7Ø5Ø          CALL    RESKFLG           ;Reset Break,Enter,Pause
283A E1         Ø7Ø6Ø          POP     HL
283B D1         Ø7Ø7Ø          POP     DE                ;Restore registers
283C C1         Ø7Ø8Ø          POP     BC
283D 3AC827     Ø7Ø9Ø FLSH6    LD      A,(CURDSK+1)      ;P/u drive #
284Ø E6Ø7       Ø71ØØ          AND     7                 ;Strip off code bits
2842 4F         Ø711Ø          LD      C,A               ;Drive # to C to
2843            Ø712Ø          @@GTDCT                   ;  get DCT vector
2843 3E51       ØØØ58          LD      A,81
2845 EF         ØØØ59          RST     4Ø
                Ø713Ø          IF      @MOD4
2846 CD6328     Ø714Ø          CALL    RSELCT            ;Get drive status in A
                Ø715Ø          ENDIF
                Ø716Ø          IF      @MOD2
                Ø717Ø          CALL    SELECT
                Ø718Ø          ENDIF
2849 Ø7         Ø719Ø          RLCA
284A Ø7         Ø72ØØ          RLCA
284B C9         Ø721Ø          RET
284C FD7EØA     Ø722Ø RESKFLG  LD      A,(IY+'K'-'A')    ;Reset 3-bit field
284F E6F8       Ø723Ø          AND     ØF8H
2851 FD77ØA     Ø724Ø          LD      (IY+'K'-'A'),A
2854 C9         Ø725Ø          RET
                Ø726Ø ;
                Ø727Ø ;        Drive disk I/O call setups
                Ø728Ø ;
2855 C5         Ø729Ø TSTDRV   PUSH    BC
2856 AF         Ø73ØØ          XOR     A                 ;Test for drive
2857 1821       Ø731Ø          JR      DIO1
2859 C5         Ø732Ø SELECT   PUSH    BC
285A 3EØ1       Ø733Ø          LD      A,1               ;Select new drive
285C 181C       Ø734Ø          JR      DIO1
285E C5         Ø735Ø RESTOR   PUSH    BC
285F 3EØ4       Ø736Ø          LD      A,4               ;Restore
2861 1817       Ø737Ø          JR      DIO1
2863 C5         Ø738Ø RSELCT   PUSH    BC
2864 3EØ7       Ø739Ø          LD      A,7               ;Reselect
```

Backup initialization

```
2866 1812       Ø74ØØ          JR      DIO1
2868 C5         Ø741Ø WRSEC    PUSH    BC
2869 3EØD       Ø742Ø          LD      A,13            ;Write sector
286B 18ØD       Ø743Ø          JR      DIO1
286D C5         Ø744Ø WRSYS    PUSH    BC
286E 3EØE       Ø745Ø          LD      A,14            ;Write protected
287Ø 18Ø8       Ø746Ø          JR      DIO1
2872 C5         Ø747Ø RDSEC    PUSH    BC
2873 3EØ9       Ø748Ø          LD      A,9             ;Read sector
2875 18Ø3       Ø749Ø          JR      DIO1
                Ø75ØØ ;
                Ø751Ø          IF      @MOD2
                Ø752Ø FMTCYL   PUSH    BC              ;Save
                Ø753Ø          LD      A,15            ;I/O command
                Ø754Ø          JR      DIO1            ;Continue
                Ø755Ø          ENDIF
                Ø756Ø ;
2877 C5         Ø757Ø VERSEC   PUSH    BC
2878 3EØA       Ø758Ø          LD      A,1Ø            ;Verify sector
287A C628       Ø759Ø DIO1     ADD     A,4Ø            ;Adjust for SVC
287C 47         Ø76ØØ          LD      B,A             ;Save tempy
287D 3AC827     Ø761Ø          LD      A,(CURDSK+1)    ;Get drive number
288Ø E6Ø7       Ø762Ø          AND     7               ;Strip diskette type bit
2882 4F         Ø763Ø          LD      C,A             ;Load up drive register
2883 78         Ø764Ø          LD      A,B             ;Get back SVC #
                Ø765Ø          IF      @MOD4
2884 F3         Ø766Ø          DI                      ;Interrupts off
                Ø767Ø          ENDIF
2885 EF         Ø768Ø          RST     4Ø
                Ø769Ø          IF      @MOD4
2886 FB         Ø77ØØ          EI                      ;Interrupts on
                Ø771Ø          ENDIF
2887 C1         Ø772Ø          POP     BC
2888 C9         Ø773Ø          RET
                Ø774Ø ;
                Ø775Ø ;        Check for correct disk
                Ø776Ø ;
2889 D5         Ø777Ø CKSWDD   PUSH    DE              ;Save DE,BC
288A C5         Ø778Ø          PUSH    BC
288B 3AØE27     Ø779Ø          LD      A,(SRCDRV$+1)   ;Get drive
288E 21C827     Ø78ØØ          LD      HL,CURDSK+1
2891 4E         Ø781Ø          LD      C,(HL)          ;Get current drive
2892 77         Ø782Ø          LD      (HL),A          ;Make curdsk our dsk
2893 2A1626     Ø783Ø          LD      HL,(BUFFER$)    ;I/O buffer
2896 11Ø2ØØ     Ø784Ø          LD      DE,2            ;Trk Ø, sect 2
2897            Ø785Ø PROTSEC  EQU     $-2
2899 CD7228     Ø786Ø          CALL    RDSEC           ;Read SIS sector
289C 2Ø1C       Ø787Ø          JR      NZ,EX2          ;Quit on read error
289E 2EC6       Ø788Ø          LD      L,ØC6H          ;Set buffer posn
28AØ 3EØØ       Ø789Ø          LD      A,$-$           ;Get original id byte
28A1            Ø79ØØ SVCTR    EQU     $-1
28A2 BE         Ø791Ø          CP      (HL)            ;Is it the same disk?
28A3 2ØØF       Ø792Ø          JR      NZ,EX1          ;NZ=error exit
28A5 3C         Ø793Ø          INC     A
28A6 28ØC       Ø794Ø          JR      Z,EX1
28A8 3D         Ø795Ø          DEC     A               ;If id byte Ø,
28A9 28Ø9       Ø796Ø          JR      Z,EX1           ; no modifying needed
28AB 3D         Ø797Ø          DEC     A               ; else dec remaining
28AC 2ØØ1       Ø798Ø          JR      NZ,$+3          ;If now Ø, make FFH
```

Backup initialization

```
28AE 3D         07990           DEC     A
28AF 77         08000           LD      (HL),A          ;Store the new id
                08010           IF      @MOD2
                08020           LD      L,0             ;Reset buffer
                08030           ENDIF
                08040           IF      @MOD4
28B0 6A         08050           LD      L,D             ;Reset buffer
                08060           ENDIF
28B1 CD6828     08070           CALL    WRSEC           ;Put it back, ck error later
28B4 79         08080 EX1       LD      A,C
28B5 32C827     08090           LD      (CURDSK+1),A    ;Restor orig drv #
28B8 C1         08100           POP     BC
28B9 D1         08110           POP     DE
28BA 211F2A     08120 EX2       LD      HL,CANTBU$      ;Go if was write error
28BD C8         08130           RET     Z
28BE C3AF26     08140           JP      EXIT4
                08150 ;
                08160 ;
                08170 ;         Message area
                08180 ;
28C1 0A         08190 DSTWP$  DB        LF,'Destination disk is write '
     44 65 73 74 69 6E 61 74
     69 6F 6E 20 64 69 73 6B
     20 69 73 20 77 72 69 74
     65 20
28DC 70         08200           DB      'protected',CR
     72 6F 74 65 63 74 65 64
     0D
28E6 49         08210 BADMPW$ DB        'Invalid master password',CR
     6E 76 61 6C 69 64 20 6D
     61 73 74 65 72 20 70 61
     73 73 77 6F 72 64 0D
28FE 1D         08220 PMTSYS$ DB        29,30,'Insert SYSTEM disk  <ENTER>',3
     1E 49 6E 73 65 72 74 20
     53 59 53 54 45 4D 20 64
     69 73 6B 20 20 3C 45 4E
     54 45 52 3E 03
291C 1D         08230 PMTSRC$ DB        29,30,'Insert SOURCE disk  <ENTER>',3
     1E 49 6E 73 65 72 74 20
     53 4F 55 52 43 45 20 64
     69 73 6B 20 20 3C 45 4E
     54 45 52 3E 03
293A 1D         08240 PMTDST$ DB        29,30,'Insert DESTINATION disk  '
     1E 49 6E 73 65 72 74 20
     44 45 53 54 49 4E 41 54
     49 4F 4E 20 64 69 73 6B
     20 20
2955 3C         08250           DB      '<ENTER>',3
     45 4E 54 45 52 3E 03
295D 1D         08260 DIFSRC$ DB        29,30,'* A L E R T *  That',27H
     1E 2A 20 41 20 4C 20 45
     20 52 20 54 20 2A 20 20
     54 68 61 74 27
2973 73         08270           DB      's not the same source disk ',CR
     20 6E 6F 74 20 74 68 65
     20 73 61 6D 65 20 73 6F
     75 72 63 65 20 64 69 73
     6B 20 0D
298F 1D         08280 DIFDST$ DB        29,30,'* A L E R T *  That',27H
```

Backup initialization

```
     1E 2A 2Ø 41 2Ø 4C 2Ø 45
     2Ø 52 2Ø 54 2Ø 2A 2Ø 2Ø
     54 68 61 74 27
29A5 73          Ø829Ø          DB      's not the same destination disk ',CR
     2Ø 6E 6F 74 2Ø 74 68 65
     2Ø 73 61 6D 65 2Ø 64 65
     73 74 69 6E 61 74 69 6F
     6E 2Ø 64 69 73 6B 2Ø ØD
29C6 53          Ø83ØØ CCMOD$   DB      'Source disk is write protected; '
     6F 75 72 63 65 2Ø 64 69
     73 6B 2Ø 69 73 2Ø 77 72
     69 74 65 2Ø 7Ø 72 6F 74
     65 63 74 65 64 3B 2Ø
29E6 4D          Ø831Ø          DB      'MOD flags not updated',CR
     4F 44 2Ø 66 6C 61 67 73
     2Ø 6E 6F 74 2Ø 75 7Ø 64
     61 74 65 64 ØD
29FC ØA          Ø832Ø BUCAO$   DB      LF,'Backup complete',CR
     42 61 63 6B 75 7Ø 2Ø 63
     6F 6D 7Ø 6C 65 74 65 ØD
2AØD ØA          Ø833Ø ABRTBU$ DB       LF,'Command aborted',14,CR
     43 6F 6D 6D 61 6E 64 2Ø
     61 62 6F 72 74 65 64 ØE
     ØD
2A1F 43          Ø834Ø CANTBU$ DB       'Can''t Backup - source disk write protected',LF
     61 6E 27 74 2Ø 42 61 63
     6B 75 7Ø 2Ø 2D 2Ø 73 6F
     75 72 63 65 2Ø 64 69 73
     6B 2Ø 77 72 69 74 65 2Ø
     7Ø 72 6F 74 65 63 74 65
     64 ØA
2A4A 44          Ø835Ø PROT$    DB      'Disk contains protected files ',CR
     69 73 6B 2Ø 63 6F 6E 74
     61 69 6E 73 2Ø 7Ø 72 6F
     74 65 63 74 65 64 2Ø 66
     69 6C 65 73 2Ø ØD
2A69             Ø836Ø BUCORE$ DEFL     $
2BØØ             Ø837Ø          ORG     $<-8+1<+8
Ø1ØØ             Ø838Ø BUF1$    DS      256
Ø1ØØ             Ø839Ø BUF2$    DS      256
Ø1ØØ             Ø84ØØ BUF3$    DS      256
```

Backup initialization

```
               Ø842Ø ;
               Ø843Ø ;
               Ø844Ø ;        Backup entry point
               Ø845Ø ;
               Ø846Ø ;
               Ø847Ø BACKUP
2EØØ           Ø848Ø          @@CKBRKC
2EØØ 3E6A      ØØØ6Ø          LD      A,1Ø6
2EØ2 EF        ØØØ61          RST     4Ø
2EØ3 28Ø4      Ø849Ø          JR      Z,BACKUPA       ;Go ahead if no break
2EØ5 21FFFF    Ø85ØØ          LD      HL,-1           ; else abort
2EØ8 C9        Ø851Ø          RET
               Ø852Ø ;
2EØ9 ED73BE26  Ø853Ø BACKUPA  LD      (SPSAV+1),SP    ;Save current SP
2EØD E5        Ø854Ø          PUSH    HL              ;Save cmdbuf
2EØE           Ø855Ø          @@BREAK Ø               ;Remove any BREAK vector
               ØØØ62          IFEQ    Ø1H,1
2EØE 21ØØØØ    ØØØ63          LD      HL,Ø
               ØØØ64          ENDIF
2E11 3E67      ØØØ65          LD      A,1Ø3
2E13 EF        ØØØ66          RST     4Ø
2E14           Ø856Ø          @@DSPLY HELLO$          ;Welcome
               ØØØ67          IFEQ    Ø1H,1
2E14 21F942    ØØØ68          LD      HL,HELLO$
               ØØØ69          ENDIF
2E17 3EØA      ØØØ7Ø          LD      A,1Ø
2E19 EF        ØØØ71          RST     4Ø
2E1A           Ø857Ø          @@FLAGS                 ;IY => flag table
2E1A 3E65      ØØØ72          LD      A,1Ø1
2E1C EF        ØØØ73          RST     4Ø
2E1D CD4C28    Ø858Ø          CALL    RESKFLG         ;Reset KFLAG bits
2E2Ø FDCBØ24E  Ø859Ø          BIT     1,(IY+'C'-'A')  ;Check on CMNDR active
2E24 217E43    Ø86ØØ          LD      HL,LDOS$
2E27 C2AF26    Ø861Ø          JP      NZ,EXIT4        ;  and exit if so
2E2A E1        Ø862Ø          POP     HL
2E2B 7E        Ø863Ø BCK1     LD      A,(HL)          ;Bypass cmdline spaces
2E2C 23        Ø864Ø          INC     HL
2E2D FE2Ø      Ø865Ø          CP      ' '
2E2F 28FA      Ø866Ø          JR      Z,BCK1
               Ø867Ø ;
               Ø868Ø ;        Scan for source partial spec
               Ø869Ø ;
2E31 11Ø226    Ø87ØØ          LD      DE,SPCFLD$      ;Pt to filespec field
2E34 Ø6Ø8      Ø871Ø          LD      B,8             ;Init for file name
2E36 FE2D      Ø872Ø          CP      '-'             ;Exclude matches?
2E38 2ØØ5      Ø873Ø          JR      NZ,BCK2         ;If '-', set flag
2E3A 32ØD26    Ø874Ø          LD      (MFLG$),A
2E3D 7E        Ø875Ø          LD      A,(HL)          ;Get next char
2E3E 23        Ø876Ø          INC     HL
2E3F CDFØ3Ø    Ø877Ø BCK2     CALL    PRSPEC          ;Parse possible filename
2E42 FE2F      Ø878Ø          CP      '/'             ;File ext?
2E44 2ØØA      Ø879Ø          JR      NZ,BCK3
2E46 11ØA26    Ø88ØØ          LD      DE,SPCFLD$+8    ;Reposn buffer ptr
2E49 Ø6Ø3      Ø881Ø          LD      B,3             ;Init for 3 chars
2E4B 7E        Ø882Ø          LD      A,(HL)
2E4C 23        Ø883Ø          INC     HL              ;Bypass the /
2E4D CDFØ3Ø    Ø884Ø          CALL    PRSPEC          ;Parse extension
               Ø885Ø ;
               Ø886Ø ;        Determine source & destination drives
               Ø887Ø ;
```

Backup initialization

```
2E5Ø FE3A      Ø888Ø BCK3    CP      ':'             ;Drive number coming?
2E52 2817      Ø889Ø         JR      Z,BCK4          ;Go if so
2E54 2B        Ø89ØØ         DEC     HL              ;Save possible parms
2E55 E5        Ø891Ø         PUSH    HL
2E56           Ø892Ø         @@DSPLY SRCNUM$         ;No drives enter, so
               ØØØ74         IFEQ    Ø1H,1
2E56 21B343    ØØØ75         LD      HL,SRCNUM$
               ØØØ76         ENDIF
2E59 3EØA      ØØØ77         LD      A,1Ø
2E5B EF        ØØØ78         RST     4Ø
2E5C 215826    Ø893Ø         LD      HL,LILBUF$      ;  prompt for them
2E5F Ø1ØØØ1    Ø894Ø         LD      BC,1<8          ;1 char response
2E62           Ø895Ø         @@KEYIN
2E62 3EØ9      ØØØ79         LD      A,9
2E64 EF        ØØØ8Ø         RST     4Ø
2E65 DAAC26    Ø896Ø         JP      C,ABRTBU        ;Quit on Break
2E68 7E        Ø897Ø         LD      A,(HL)          ;Get response. Restore
2E69 E1        Ø898Ø         POP     HL              ;  command buffer. Ignore
2E6A DA        Ø899Ø         DB      ØDAH            ;  next 2 inst with JP C,
2E6B 7E        Ø9ØØØ BCK4    LD      A,(HL)          ;P/u source drive #
2E6C 23        Ø9Ø1Ø         INC     HL              ;Bump to separator
2E6D D63Ø      Ø9Ø2Ø         SUB     'Ø'             ;Adj to binary
2E6F FEØ8      Ø9Ø3Ø         CP      8               ;Error if not in
2E71 D29526    Ø9Ø4Ø         JP      NC,EXIT2        ;  the range <Ø-7>
2E74 32ØE27    Ø9Ø5Ø         LD      (SRCDRV$+1),A   ;Stuff source drive
2E77 7E        Ø9Ø6Ø BCK5    LD      A,(HL)          ;P/u char or separator
2E78 23        Ø9Ø7Ø         INC     HL              ;Bump ptr
2E79 FE3A      Ø9Ø8Ø         CP      ':'             ;Find dest drive?
2E7B 281F      Ø9Ø9Ø         JR      Z,BCK6          ;Get drive # if :
2E7D FE3Ø      Ø91ØØ         CP      3ØH             ; let prepositions thru
2E7F 3ØF6      Ø911Ø         JR      NC,BCK5
2E81 FE2Ø      Ø912Ø         CP      2ØH             ;Or a space separator
2E83 28F2      Ø913Ø         JR      Z,BCK5
2E85 2B        Ø914Ø         DEC     HL              ;Save possible parms
2E86 E5        Ø915Ø         PUSH    HL
2E87           Ø916Ø         @@DSPLY DSTNUM$         ;Prompt for dest drive
               ØØØ81         IFEQ    Ø1H,1
2E87 21D143    ØØØ82         LD      HL,DSTNUM$
               ØØØ83         ENDIF
2E8A 3EØA      ØØØ84         LD      A,1Ø
2E8C EF        ØØØ85         RST     4Ø
2E8D 215826    Ø917Ø         LD      HL,LILBUF$      ;Use for keyin buffer
2E9Ø Ø1ØØØ1    Ø918Ø         LD      BC,1<8          ;1 char only
2E93           Ø919Ø         @@KEYIN
2E93 3EØ9      ØØØ86         LD      A,9
2E95 EF        ØØØ87         RST     4Ø
2E96 DAAC26    Ø92ØØ         JP      C,ABRTBU        ;Quit on Break
2E99 7E        Ø921Ø         LD      A,(HL)          ;Get response. Restore
2E9A E1        Ø922Ø         POP     HL              ;  buffer. Ignore next 2
2E9B DA        Ø923Ø         DB      ØDAH            ;  inst with JP C,nn
2E9C 7E        Ø924Ø BCK6    LD      A,(HL)          ;P/u dest drive #
2E9D 23        Ø925Ø         INC     HL              ;Bump line ptr
2E9E D63Ø      Ø926Ø         SUB     'Ø'             ;Adjust to binary
2EAØ FEØ8      Ø927Ø         CP      8               ;Error if not in the
2EA2 D29526    Ø928Ø         JP      NC,EXIT2        ;  range <Ø-7>
2EA5 327B27    Ø929Ø         LD      (DSTDRV$+1),A   ;Stuff dest drive
               Ø93ØØ ;
2EA8 115442    Ø931Ø         LD      DE,PRMTBL$      ;P/u parm table ptr
2EAB D5        Ø932Ø         PUSH    DE              ;Also in IX to check
```

Backup initialization

```
2EAC DDE1       09330          POP      IX                ;  responses
2EAE            09340          @@PARAM                    ;Get parms if any
2EAE 3E11       00088          LD       A,17
2EB0 EF         00089          RST      40
2EB1 21A343     09350          LD       HL,PRMERR$        ;Init "parm error
2EB4 2005       09360          JR       NZ,$EX4           ;Quit on parm error
2EB6 DD7E30     09370          LD       A,(IX+DATRSP)     ;Date can only be STR
2EB9 E6C0       09380          AND      VAL!SW            ;This must be string
2EBB C2AF26     09390 $EX4     JP       NZ,EXIT4          ;Quit if not
                09400 ;
                09410 ;        Check on Source = Destination
                09420 ;
2EBE 3A0E27     09430          LD       A,(SRCDRV$+1)     ;P/u source drive
2EC1 217B27     09440          LD       HL,DSTDRV$+1
2EC4 AE         09450          XOR      (HL)              ;Match against dest
2EC5 32CF27     09460          LD       (SXORD+1),A       ;0 if S=D, <>0 if S<>D
2EC8 200D       09470          JR       NZ,DATPRM         ;Bypass if source <> dest
2ECA            09480          @@FLAGS                    ;Else test if <DO> proc
2ECA 3E65       00090          LD       A,101
2ECC EF         00091          RST      40
2ECD FDCB126E   09490          BIT      5,(IY+'S'-'A')
2ED1 219642     09500          LD       HL,NOINDO$        ;"can't do single...
2ED4 C2AF26     09510          JP       NZ,EXIT4          ;Abort if from <DO>
                09520 ;
                09530 ;        Check on date entries
                09540 ;
2ED7 210000     09550 DATPRM   LD       HL,0              ;P/u date="from-to"
2EDA 7C         09560          LD       A,H
2EDB B5         09570          OR       L
2EDC 282F       09580          JR       Z,CKCLAS          ;Bypass if not entered
2EDE 7E         09590          LD       A,(HL)            ;Check for "-to"
2EDF FE2D       09600          CP       '-'
2EE1 2815       09610          JR       Z,CKTO            ;Go if no From used
2EE3 3E80       09620          LD       A,80H             ;Set From bit
2EE5 320126     09630          LD       (FTFLG$),A        ;Note From entered
2EE8 CD0B31     09640          CALL     PAKDAT            ;Pack the date entry
2EEB ED438026   09650          LD       (FMPAKD$),BC      ;Save From packed date
2EEF 7E         09660          LD       A,(HL)            ;Ck if more in date parm
2EF0 FE22       09670          CP       '"'               ;End of string?
2EF2 280D       09680          JR       Z,FRCDAT          ;Go if so
2EF4 FE2D       09690          CP       '-'               ;Check for "-to"
2EF6 2015       09700          JR       NZ,CKCLAS         ;Done if not
2EF8 23         09710 CKTO     INC      HL                ;Bypass the '-'
2EF9 7E         09720          LD       A,(HL)            ;Ck for end of parm
2EFA FE22       09730          CP       '"'
2EFC 280F       09740          JR       Z,CKCLAS          ;Go if done
2EFE CD0B31     09750          CALL     PAKDAT            ;Pack To date
2F01 3A0126     09760 FRCDAT   LD       A,(FTFLG$)        ;P/u From/To flag and
2F04 F601       09770          OR       1                 ;  set To bit
2F06 320126     09780          LD       (FTFLG$),A
2F09 ED438226   09790          LD       (TOPAKD$),BC      ;Save To packed date
                09800 ;
                09810 ;        Check on parms to force CLASS backup
                09820 ;
2F0D 0600       09830 CKCLAS   LD       B,0               ;Init class flag
2F0F 110000     09840 SYSPRM   LD       DE,0              ;SYS parm used?
2F12 7A         09850          LD       A,D
2F13 B3         09860          OR       E
2F14 2802       09870          JR       Z,INVPRM          ;Go if not
```

Backup initialization

```
2F16 CBFØ              SET     6,B             ;Set 6 if SYS
2F18 11ØØØØ     INVPRM LD      DE,Ø            ;INV parm used?
2F1B 7A                LD      A,D
2F1C B3                OR      E
2F1D 28Ø2              JR      Z,CKCLA1        ;Go if not
2F1F CBD8              SET     3,B             ;Set 3 if INV
2F21 78         CKCLA1 LD      A,B
2F22 328426            LD      (CLSFLG$),A     ;Store by class flag
2F25 3AØ226            LD      A,(SPCFLD$)     ;Get 1st char of possible
2F28 D62Ø              SUB     ' '             ;  file name
2F2A 47                LD      B,A             ;Save test result and
2F2B 3AØA26            LD      A,(SPCFLD$+8)   ;  check if extension used
2F2E D62Ø              SUB     ' '             ;Ck for ext
2F3Ø BØ                OR      B               ;A <> Ø if partspec
2F31 47                LD      B,A             ;Hold in reg B
                ;
                ;      Merge all "CLASS" parms together
                ;
2F32 DD7EØC            LD      A,(IX+SYSRSP)   ;System files
2F35 DDB613            OR      (IX+INVRSP)     ;Invisible files
2F38 DDB61A            OR      (IX+MODRSP)     ;Mod flag files
2F3B DDB637            OR      (IX+NEWRSP)     ;Files not on dest
2F3E DDB63E            OR      (IX+OLDRSP)     ;Files on dest
2F41 DDB623            OR      (IX+QRSP)       ;Query forces by class
2F44 4F                LD      C,A             ;Hold value
2F45 E6AØ              AND     VAL!STR         ;Above parms only SWITCH
2F47 21A343            LD      HL,PRMERR$      ;Init "parm error
2F4A C2AF26            JP      NZ,EXIT4        ;Quit if not switches only
2F4D B1                OR      C
2F4E BØ                OR      B               ;Merge with partspec
2F4F DDB63Ø            OR      (IX+DATRSP)     ;D=" mm/dd/yy-mm/dd/yy"
                ;
                ;      Advise backup by class if any class parameter
                ;
2F52 32Ø141            LD      (CLSTST+1),A    ;Set for all flags
2F55 28Ø6              JR      Z,GETDAT        ;Z=may be mirror image
2F57                   @@LOGOT CLASS$          ;  else log by class msg
                       IFEQ    Ø1H,1
2F57 21Ø344            LD      HL,CLASS$
                       ENDIF
2F5A 3EØC              LD      A,12
2F5C EF                RST     4Ø
                ;
                ;      Recover today's date
                ;
2F5D 217826     GETDAT LD      HL,DATFLD$      ;Date storage buffer
2F6Ø                   @@DATE                  ;Get date
2F6Ø 3E12              LD      A,18
2F62 EF                RST     4Ø
2F63 1A                LD      A,(DE)          ;Check if date in system
2F64 B7                OR      A
2F65 2ØØ6              JR      NZ,GETGM        ;Go if it is
2F67 21EF43            LD      HL,NODAT$
2F6A                   @@LOGOT                 ;Show "no date" if none
                       IFEQ    ØØH,1
                       LD      HL,
                       ENDIF
2F6A 3EØC              LD      A,12
2F6C EF                RST     4Ø
```

Backup initialization

```
2F6D D5           GETGM    PUSH    DE                ;Save date$
2F6E E5                    PUSH    HL                ;  and date buffer
2F6F 114144                LD      DE,RES$           ;See if SYS modules resident
2F72                       @@GTMOD                   ;  in case needed later
2F72 3E53                  LD      A,83
2F74 EF                    RST     4Ø
2F75 2ØØ4                  JR      NZ,GETDAT1        ;Skip if none res'ed
2F77 ED532941              LD      (RESLOC+1),DE     ;Store the module loc
                  ;
                  ;        Get SYS2 loaded for password hash
                  ;
2F7B E1           GETDAT1  POP     HL
2F7C D1                    POP     DE
2F7D CD6A41                CALL    GETSYS2           ;Get sys2 and move date
                  ;
                  ;        Check on (X) parm for source/dest swap
                  ;
2F8Ø 3AØE27                LD      A,(SRCDRV$+1)     ;If source is not Ø,
2F83 B7                    OR      A                 ; then let PMTSRC handle
2F84 2ØØF                  JR      NZ,SRCDFT
2F86 F68Ø                  OR      8ØH               ;Set to SRC code
2F88 4F                    LD      C,A               ;Save if needed
2F89 3ACA26                LD      A,(XPARM$+1)      ;Source is drive Ø,
2F8C 3C                    INC     A                 ; if (X), then swap
2F8D F5                    PUSH    AF
2F8E 211C29                LD      HL,PMTSRC$
2F91 CCC127                CALL    Z,FRCPMT          ;Force prompt on (X)
2F94 F1                    POP     AF
2F95 C4ØD27       SRCDFT   CALL    NZ,SRCDRV$        ;Prompt for source
2F98 CD5E28                CALL    RESTOR            ;Get set to see if a
2F9B CDBF41                CALL    CKDRV             ; source disk mounted
2F9E 28ØA                  JR      Z,GOTSRC          ;Z=ok
2FAØ F5                    PUSH    AF
2FA1 211C29                LD      HL,PMTSRC$        ;Else prompt "Insert...
2FA4 CDC127                CALL    FRCPMT
2FA7 F1                    POP     AF
2FA8 18EB                  JR      SRCDFT            ; and then check again
                  ;
                  ;        Get source disk attributes
                  ;
2FAA FD7EØ3       GOTSRC   LD      A,(IY+3)          ;P/u 5" or 8" from
2FAD E62Ø                  AND     2ØH               ;  DCT+3, bit 5
2FAF 324D3Ø                LD      (TST5_8+1),A      ;  and save for later
2FB2 CD5528                CALL    TSTDRV            ;Ck for active DCT
2FB5 C29726                JP      NZ,EXIT3          ;  and quit if not
2FB8 21ØØ2D                LD      HL,BUF3$          ;Disk buffer
                  ;
                           IF      @MOD2
                           CALL    GETPSEC           ;Get prot sector
                           JP      NZ,EXIT3          ;Go on error
                           CP      6                 ;Directory?
                           JP      NZ,DIRERR         ;Nope, go!
                           ENDIF
2FBB 11ØØØØ                LD      DE,Ø              ;Set to track/sector Ø/Ø
2FBE CD7228                CALL    RDSEC             ;Read boot
2FC1 C29726                JP      NZ,EXIT3          ;Quit on read error
2FC4 3AØ22D                LD      A,(BUF3$+2)       ;P/u dir track
2FC7 FD77Ø9                LD      (IY+9),A          ;  & stuff in table
                           IF      @MOD2
```

Backup initialization

```
                         LD      DE,(PROTSEC)    ;Get info sector
                         ENDIF
                         IF      @MOD4
2FCA 1C                  INC     E               ;Point to SYSINFO sector
2FCB 1C                  INC     E
                         ENDIF
2FCC 262B                LD      H,BUF1$<-8      ;Use this disk buffer
2FCE CD7228              CALL    RDSEC           ;Read the info sector
2FD1 C29726              JP      NZ,EXIT3        ;Quit on read error
2FD4 3AC62B              LD      A,(BUF1$+ØC6H)  ;Get & save id byte
2FD7 32A128              LD      (SVCTR),A
2FDA 3C                  INC     A
2FDB 2815                JR      Z,CKGAT
               ;
               ;         Check write protect status
               ;
2FDD 3D                  DEC     A               ;Need to check?
2FDE 2812                JR      Z,CKGAT         ;Go if not
2FEØ CD5E28              CALL    RESTOR          ;Start the drive
2FE3 CD6328              CALL    RSELCT          ;Ck if WP
2FE6 FDB6Ø3              OR      (IY+3)          ;Merge in soft WP
2FE9 Ø7                  RLCA                    ;Push WP to CF
2FEA 3ØØ6                JR      NC,CKGAT        ;Bypass if not WP
2FEC 211F2A    CANTBU    LD      HL,CANTBU$
2FEF C3AF26              JP      EXIT4
               ;
2FF2 FD56Ø9    CKGAT     LD      D,(IY+9)        ;Directory track,
2FF5 1EØØ                LD      E,Ø             ;  sector Ø
2FF7 21ØØ2B              LD      HL,BUF1$
2FFA CD7228              CALL    RDSEC           ;Read GAT
2FFD FEØ6                CP      6               ;Ensure directory cyl
2FFF C29226              JP      NZ,DIRERR       ;Quit on any other error
3ØØ2 CDCF3Ø              CALL    TSTMPW          ;Get password if needed
               ;
               ;         Check if destination formatted & not protected
               ;
3ØØ5 3A7B27              LD      A,(DSTDRV$+1)   ;If dest is not Ø,
3ØØ8 B7                  OR      A               ; then let DSTDRV handle
3ØØ9 2ØØF                JR      NZ,DSTDFT
3ØØB F64Ø                OR      4ØH             ;Set DST code
3ØØD 4F                  LD      C,A             ;Save if needed
3ØØE 3ACA26              LD      A,(XPARM$+1)    ;Dest is drive Ø
3Ø11 3C                  INC     A               ;If (X), then swap
3Ø12 F5                  PUSH    AF
3Ø13 213A29              LD      HL,PMTDST$
3Ø16 CCC127              CALL    Z,FRCPMT        ;Force prompt on (X)
3Ø19 F1                  POP     AF
3Ø1A C47A27    DSTDFT    CALL    NZ,DSTDRV$      ;Get dest drive
3Ø1D CD5E28              CALL    RESTOR          ;Restore destination
3Ø2Ø CD6328              CALL    RSELCT          ;Test it
3Ø23 2Ø18                JR      NZ,PMTDD        ;Might be signal from
                                                 ;HD driver
3Ø25 Ø7                  RLCA
3Ø26 FDB6Ø3              OR      (IY+3)          ;Merge in soft WP
3Ø29 CB7F                BIT     7,A             ; Check on WP status
3Ø2B 21C128              LD      HL,DSTWP$       ;Dest write prot...
3Ø2E C2AF26              JP      NZ,EXIT4        ;Jp if write protected
                                                 ;If hard drive, don't
3Ø31 FDCBØ35E            BIT     3,(IY+3)        ;  try to test for write
```

Backup initialization

```
3035 C2AE30    11510          JP      NZ,RECON          ;  but go to re-construct
3038 CDBF41    11520          CALL    CKDRV             ;Ck if diskette in place
303B 280A      11530          JR      Z,GOTDST
303D F5        11540 PMTDD    PUSH    AF                ;Kludge a force of
303E 213A29    11550          LD      HL,PMTDST$
3041 CDC127    11560          CALL    FRCPMT
3044 F1        11570          POP     AF
3045 18D3      11580          JR      DSTDFT            ;  the destination prompt
               11590 ;
               11600 ;        Check 5" vs 8" for forced reconstruction
               11610 ;
3047 FD7E03    11620 GOTDST   LD      A,(IY+3)
304A E620      11630          AND     20H               ;See if 5/8 mismatch
304C EE00      11640 TST5_8   XOR     0                 ;P/u source size
304E C2AE30    11650          JP      NZ,RECON          ;Go if different
3051 110000    11660          LD      DE,0
3054 CD7728    11670          CALL    VERSEC            ;Verify boot sector readable
3057 2806      11680          JR      Z,CKDST           ;Jump if ok
               11690 ;
               11700 ;        Destination not formatted, abort
               11710 ;
3059 21C942    11720          LD      HL,NOFMT$         ;Init "Not formatted
305C C3AF26    11730          JP      EXIT4             ;Display and abort
               11740 ;
               11750 ;        Check destination attributes
               11760 ;
305F 21002D    11770 CKDST    LD      HL,BUF3$
3062 110000    11780          LD      DE,0              ;SET for track/sector 0/0
3065 CD7228    11790          CALL    RDSEC             ;Read dest boot
3068 C29726    11800          JP      NZ,EXIT3
306B 3A022D    11810          LD      A,(BUF3$+2)       ;P/u its dir track
306E 57        11820          LD      D,A               ;Set up in D
306F 21002C    11830          LD      HL,BUF2$
3072 5D        11840          LD      E,L               ;  and 0 in E
3073 CD7228    11850          CALL    RDSEC             ;Read dest GAT
3076 FE06      11860          CP      6                 ;Ensure a dir cyl
3078 C29226    11870          JP      NZ,DIRERR         ;Quit on any other error
307B 2ACC2B    11880          LD      HL,(BUF1$+TKCAP)         ;P/u source capacity
307E ED5BCC2C  11890          LD      DE,(BUF2$+TKCAP)         ;P/u dest capacity
3082 3AA128    11900          LD      A,(SVCTR)         ;If id byte was FF
3085 3C        11910          INC     A
3086 2807      11920          JR      Z,SHOPROT         ; then force recon
3088 3D        11930          DEC     A                 ;If id was not 0
3089 200C      11940          JR      NZ,TSTCAP         ;  then test sizes
308B CB64      11950          BIT     4,H               ;If types differ
308D 2808      11960          JR      Z,TSTCAP          ;  force reconstruct
               11970 ;
308F           11980 SHOPROT @@LOGOT PROT$              ;Show reconstruct invoked
               00106          IFEQ    01H,1
308F 214A2A    00107          LD      HL,PROT$
               00108          ENDIF
3092 3E0C      00109          LD      A,12
3094 EF        00110          RST     40
3095 1817      11990          JR      RECON             ;Skip next tests
               12000 ;
3097 7C        12010 TSTCAP   LD      A,H               ;Den/sides match?
3098 AA        12020          XOR     D                 ;Force Reconstruct if
3099 E660      12030          AND     60H               ;  density & sides
309B 2011      12040          JR      NZ,RECON          ;  differ
```

Backup initialization

```
3Ø9D 7D         12Ø5Ø           LD      A,L             ;Test # of cyls
3Ø9E 93         12Ø6Ø           SUB     E
3Ø9F 28ØA       12Ø7Ø           JR      Z,BYCLAS        ;Jump if same
                12Ø8Ø ;
                12Ø9Ø ;         Cylinder count differs - question Mirror
                121ØØ ;
3ØA1 3AØ141     1211Ø           LD      A,(CLSTST+1)    ;But don't question if
3ØA4 B7         1212Ø           OR      A               ; Class parms already
3ØA5 C2ØØ41     1213Ø           JP      NZ,CLSTST       ; entered
3ØA8 CDB73Ø     1214Ø           CALL    MIRROR          ;Attempt mirror?
3ØAB CAØØ41     1215Ø BYCLAS    JP      Z,CLSTST        ;Jump if mirror to be tried
3ØAE            1216Ø RECON     @@LOGOT RECON$          ;"backup re-con...
                ØØ111           IFEQ    Ø1H,1
3ØAE 218Ø44     ØØ112           LD      HL,RECON$
                ØØ113           ENDIF
3ØB1 3EØC       ØØ114           LD      A,12
3ØB3 EF         ØØ115           RST     4Ø
3ØB4 C31341     1217Ø           JP      MVBYCLS         ;Go do file backup
                1218Ø ;
                1219Ø ;         Different # of tracks - Prompt for mirror
                122ØØ ;
3ØB7            1221Ø MIRROR    @@DSPLY MIRROR$         ;"Attempt mirror...
                ØØ116           IFEQ    Ø1H,1
3ØB7 219B44     ØØ117           LD      HL,MIRROR$
                ØØ118           ENDIF
3ØBA 3EØA       ØØ119           LD      A,1Ø
3ØBC EF         ØØ12Ø           RST     4Ø
3ØBD 215926     1222Ø           LD      HL,LILBUF$+1    ;Keyin buffer
3ØCØ Ø1ØØØ3     1223Ø QM1       LD      BC,3<8          ;3 chars max
3ØC3            1224Ø           @@KEYIN
3ØC3 3EØ9       ØØ121           LD      A,9
3ØC5 EF         ØØ122           RST     4Ø
3ØC6 DAAC26     1225Ø           JP      C,ABRTBU        ;Quit on break
3ØC9 7E         1226Ø           LD      A,(HL)
3ØCA CBAF       1227Ø           RES     5,A             ;Convert to UC
3ØCC FE59       1228Ø           CP      'Y'             ;Ret Z if Yes
3ØCE C9         1229Ø           RET
                123ØØ ;
                1231Ø ;         Get & check Disk master password
                1232Ø ;
3ØCF 2ACE2B     1233Ø TSTMPW    LD      HL,(BUF1$+PSWD) ;P/u src MPW
3ØD2 11EØ42     1234Ø           LD      DE,PASSWORD     ;If "PASSWORD",
3ØD5 AF         1235Ø           XOR     A               ;  don't prompt
3ØD6 ED52       1236Ø           SBC     HL,DE
3ØD8 C8         1237Ø           RET     Z
3ØD9 11ØØØØ     1238Ø           LD      DE,$-$          ;P/u User entry
3ØDA            1239Ø MPWPRM    EQU     $-2
3ØDC 21D344     124ØØ           LD      HL,PMTMPW$      ;Init "Enter MPW
3ØDF CD6441     1241Ø           CALL    GETMPW          ;Get the user's response
3ØE2 EB         1242Ø           EX      DE,HL
3ØE3 2ACE2B     1243Ø           LD      HL,(BUF1$+PSWD)
3ØE6 AF         1244Ø           XOR     A
3ØE7 ED52       1245Ø           SBC     HL,DE           ;Entry match?
3ØE9 C8         1246Ø           RET     Z               ;Ret if MPW match
3ØEA 21E628     1247Ø           LD      HL,BADMPW$      ;  else init "bad MPW...
3ØED C3AF26     1248Ø           JP      EXIT4           ;Don't do the backup
                1249Ø ;
                125ØØ ;         Routine to parse partial filespecs & cvrt to UC
                1251Ø ;
```

```
30F0 FE24      PRSPEC  CP      '$'             ;Wild character?
30F2 280A              JR      Z,PS1           ;Always a match
30F4 FE41              CP      'A'             ;Filename entered?
30F6 3006              JR      NC,PS1
30F8 FE3A              CP      '9'+1           ;Ck on 0-9
30FA D0                RET     NC
30FB FE30              CP      '0'
30FD D8                RET     C
30FE FE61      PS1     CP      'a'             ;Cvrt to UC if needed
3100 3802              JR      C,$+4
3102 CBAF              RES     5,A             ;Convert to upper case
3104 12                LD      (DE),A          ;Save in partspec buffer
3105 13                INC     DE              ;Bump buffer
3106 7E                LD      A,(HL)          ;Get next char and
3107 23                INC     HL              ;  bump string ptr
3108 10E6              DJNZ    PRSPEC
310A C9                RET
               ;
               ;       Pack user date string
               ;
310B 7E        PAKDAT  LD      A,(HL)
310C 0E2F              LD      C,'/'           ;Init separator
310E CD5431            CALL    PARSDAT         ;Parse entry
3111 203B              JR      NZ,BADFMT       ;Jump on format error
3113 EB                EX      DE,HL
3114 3A5826            LD      A,(LILBUF$)     ;Is year a leap year?
3117 E603              AND     3
3119 21EC44            LD      HL,MAXDAYS+1    ;Set Feb to have 29 days
311C 2001              JR      NZ,$+3          ; if so
311E 34                INC     (HL)
311F 3A5A26            LD      A,(LILBUF$+2)   ;P/u month
3122 3D                DEC     A               ;Range check
3123 FE0C              CP      12
3125 3027              JR      NC,BADFMT       ;Go if 0 or >12
3127 2B                DEC     HL              ;Point to Jan entry
3128 85                ADD     A,L             ;Index the month
3129 6F                LD      L,A
312A 7C                LD      A,H
312B CE00              ADC     A,0
312D 67                LD      H,A
312E 3A5926            LD      A,(LILBUF$+1)   ;P/u day entry
3131 3D                DEC     A               ;Reduce for test (0->FF)
3132 BE                CP      (HL)
3133 3019              JR      NC,BADFMT       ;Go if too large (or 0)
3135 215A26            LD      HL,LILBUF$+2    ;Pt to month
3138 7E                LD      A,(HL)          ;P/u month
3139 2B                DEC     HL              ;Pt to day
313A 47                LD      B,A             ;Save it
313B 7E                LD      A,(HL)          ;P/u day
313C 2B                DEC     HL              ;Pt to year
313D 07                RLCA                    ;Shift day to 3-7
313E 07                RLCA
313F 07                RLCA
3140 4F                LD      C,A
3141 7E                LD      A,(HL)          ;P/u year
3142 D650              SUB     80              ;Adjust for offset
3144 3001              JR      NC,$+3          ;If entry < 1980,
3146 AF                XOR     A               ; then use 1980
3147 0F                RRCA                    ;Shift into bits 5-7
```

Backup initialization

```
3148 0F                    RRCA
3149 0F                    RRCA
314A B0                    OR      B              ; & merge with month
314B 47                    LD      B,A
314C EB                    EX      DE,HL
314D C9                    RET
314E 21F744        BADFMT  LD      HL,BADFMT$
3151 C3AF26                JP      EXIT4
                   ;
                   ;       Routine to parse DATE/TIME entry
                   ;
3154 115A26        PARSDAT LD      DE,LILBUF$+2   ;Point to buf end
3157 0603                  LD      B,3            ;Process 3 fields
3159 D5            PRSD1   PUSH    DE             ;Save pointer
315A CD6931                CALL    PRSD2          ;Get a digit pair
315D D1                    POP     DE             ;Recover pointer
315E C0                    RET     NZ             ;Ret if bad digit pair
315F 12                    LD      (DE),A         ; else stuff the value
3160 1B                    DEC     DE             ;Backup the pointer
3161 05                    DEC     B              ;Loop countdown
3162 C8                    RET     Z
3163 7E                    LD      A,(HL)         ;Ck for valid separator
3164 23                    INC     HL             ;Bump pointer
3165 B9                    CP      C              ;Separator char required
3166 28F1                  JR      Z,PRSD1        ;Loop if match
3168 C9                    RET                    ;Else ret bad (NZ)
                   ;
                   ;       Routine to parse a digit pair
                   ;
3169 CD8031        PRSD2   CALL    PRS4           ;Get a digit
316C 3010                  JR      NC,PRSD3       ;Jump if bad digit
316E 5F                    LD      E,A            ;Multiply by ten
316F 07                    RLCA
3170 07                    RLCA
3171 83                    ADD     A,E
3172 07                    RLCA
3173 5F                    LD      E,A
3174 CD8031                CALL    PRS4           ;Get another digit
3177 3005                  JR      NC,PRSD3       ;Jump on bad digit
3179 83                    ADD     A,E            ;Accumulate new digit
317A 5F                    LD      E,A            ;Save 2-digit value
317B AF                    XOR     A              ;Clear flags
317C 7B                    LD      A,E            ;Xfer field value
317D C9                    RET
317E B7            PRSD3   OR      A              ;Set NZ
317F C9                    RET
3180 7E            PRS4    LD      A,(HL)         ;P/u a digit &
3181 23                    INC     HL             ;Convert to binary
3182 D630                  SUB     30H
3184 FE0A                  CP      10             ;Set CF if good
3186 C9                    RET
                   ;
                   ;       Save PC for later use
                   ;
3200               CORE$   DEFL    $<-8+1<8       ;Set to next page
2E00                       ORG     BACKUP         ;Set for MIRROR exec
3200               MIRBU   EQU     CORE$
2E00                       LORG    MIRBU          ;Load origin
                   ;
```

```
2EØØ          137ØØ         SUBTTL  '<Mirror Image Backup>'
```

Mirror Image Backup

```
                 ;
2EØØ             *GET    BACKUP2:3
                 ;BACKUP2/ASM - Mirror Image Backup
                 *MOD
                 ;
2EØØ CD7A27              CALL    DSTDRV$         ;Prompt for dest but
2EØ3 CD8727              CALL    PMTDST          ;  don't test yet
2EØ6 21ØØ2D              LD      HL,BUF3$
2EØ9 55                  LD      D,L             ;Set cyl to Ø
2EØA 1EØ1                LD      E,1             ;Read sector 1 for step
2EØC CD7228              CALL    RDSEC           ;Read BOOT
2EØF C29726              JP      NZ,EXIT3        ;Quit on read error
2E12 3AØØ26              LD      A,(BOOTST$)     ;P/u the boot step rate
2E15 6F                  LD      L,A
2E16 7E                  LD      A,(HL)
2E17 E6Ø3                AND     3               ;  from bits Ø-1
2E19 32Ø63Ø              LD      (BSMIR+1),A     ;Save for later
2E1C 3AØ22D              LD      A,(BUF3$+2)     ;Get dir cylinder
2E1F 57                  LD      D,A             ;  into D
2E2Ø 21ØØ2C              LD      HL,BUF2$        ;Use this buffer now
2E23 5D                  LD      E,L             ;Set sector Ø
2E24 CD7228              CALL    RDSEC           ;Read the dest GAT
2E27 FEØ6                CP      6               ;Expect error 6 here
2E29 3E14                LD      A,2Ø            ;Init "GAT read error
2E2B C29726              JP      NZ,EXIT3        ;  and abort on an error
2E2E 21CE2B              LD      HL,BUF1$+ØCEH   ;Source GAT
2E31 11CE2C              LD      DE,BUF2$+ØCEH   ;Dest GAT
2E34 Ø6ØA                LD      B,1Ø            ;Compare pack names
2E36 1A          CPRID   LD      A,(DE)          ;  and passwords
2E37 BE                  CP      (HL)
2E38 286Ø                JR      Z,IDMATCH
                 ;
                 ;       No match - move disk name into message
                 ;
2E3A 21DØ2C              LD      HL,BUF2$+ØDØH
2E3D 115A32              LD      DE,PACKID$+5    ;Move name into
2E4Ø Ø1Ø8ØØ              LD      BC,8            ;  display message field
2E43 EDBØ                LDIR
2E45 116932              LD      DE,PACKID$+2Ø   ;Move date into
2E48 ØEØ8                LD      C,8             ;  message field
2E4A EDBØ                LDIR
2E4C                     @@LOGOT DIFID$          ;"diff pack ids..
                         IFEQ    Ø1H,1
2E4C 213332              LD      HL,DIFID$
                         ENDIF
2E4F 3EØC                LD      A,12
2E51 EF                  RST     4Ø
2E52                     @@FLAGS                 ;If DOing, don't!
2E52 3E65                LD      A,1Ø1
2E54 EF                  RST     4Ø
2E55 FDCB126E            BIT     5,(IY+'S'-'A')
2E59 2Ø3C                JR      NZ,PACKNDO      ;Abort if JCL going
                 ;
                 ;       If MPW = "PASSWORD", just query Y,N
                 ;
2E5B 2ACE2C              LD      HL,(BUF2$+ØCEH) ;P/u disk MPW
2E5E 11EØ42              LD      DE,PASSWORD     ;P/u hash for "PASSWORD"
2E61 AF                  XOR     A
2E62 ED52                SBC     HL,DE           ;Does it match disk MPW?
2E64 2818                JR      Z,PMTYN         ;Go get Y or N if so
```

Mirror Image Backup

```
                  ;
                  ;       User must enter Current Pack's MPW to proceed
                  ;
2E66 217232       OLDMPW  LD      HL,OLDMPW$       ;"What's the old MPW?
2E69 110000               LD      DE,0             ;Force prompt of message
2E6C CD6441               CALL    GETMPW           ;Grab user input to match
                  ;
                  ;       Routine to test master password for match
                  ;
2E6F EB                   EX      DE,HL            ;Xfer hashed MPW to DE
2E70 2ACE2C               LD      HL,(BUF2$+0CEH) ;Grab pack MPW
2E73 AF                   XOR     A                ;Clear carry flag
2E74 ED52                 SBC     HL,DE            ;Did user enter pack MPW?
2E76 21E628               LD      HL,BADMPW$       ;Init "Bad MPW" just in case
2E79 C2AF26               JP      NZ,EXIT4         ;Abort if no match
2E7C 1820                 JR      $A1              ;PW good, continue backup
                  ;
2E7E              PMTYN   @@DSPLY PMTYN$           ;"Backup anyway?"
                          IFEQ    01H,1
2E7E 21A432               LD      HL,PMTYN$
                          ENDIF
2E81 3E0A                 LD      A,10
2E83 EF                   RST     40
2E84 215826               LD      HL,LILBUF$       ;Prompt to continue
2E87 010003               LD      BC,3<8           ;  since ID's differ
2E8A                      @@KEYIN
2E8A 3E09                 LD      A,9
2E8C EF                   RST     40
2E8D DAAC26               JP      C,ABRTBU         ;Exit on break
2E90 7E                   LD      A,(HL)
2E91 CBAF                 RES     5,A              ;Make answer upper case
2E93 FE59                 CP      'Y'              ;Was answer Yes?
2E95 2807                 JR      Z,$A1            ;Go if continue
2E97 C3AC26       PACKNDO JP      ABRTBU           ;  else abort
                  ;
2E9A 13           IDMATCH INC     DE
2E9B 23                   INC     HL
2E9C 1098                 DJNZ    CPRID
2E9E 21602C       $A1     LD      HL,BUF2$+60H     ;Dest lockout table
2EA1 11602B               LD      DE,BUF1$+60H     ;Source lockout table
2EA4 0660                 LD      B,60H            ;Init to compare 96 posns
2EA6 1A           CPRLOK  LD      A,(DE)           ;P/u lockout byte
2EA7 2F                   CPL                      ;Reset all used bits
2EA8 4F                   LD      C,A              ;  and save results
2EA9 D5                   PUSH    DE
2EAA 7B                   LD      A,E              ;Now posn to GAT byte
2EAB D660                 SUB     60H              ;  for that track
2EAD 5F                   LD      E,A
2EAE 1A                   LD      A,(DE)           ;P/u free/used
2EAF D1                   POP     DE               ;Pt back to lockout
2EB0 A1                   AND     C                ;Merge non-locked and in use
2EB1 A6                   AND     (HL)             ;That much must be free on dest
2EB2 C2BC31               JP      NZ,NOTMIR        ;  else "dest disk flawed
2EB5 13                   INC     DE
2EB6 23                   INC     HL
2EB7 10ED                 DJNZ    CPRLOK           ;Loop thru all cyls
                  ;
                  ;       Dest can take backup, insert HALT for swap test
                  ;
```

Mirror Image Backup

```
2EB9 CD8727          CALL    PMTDST          ;Prompt dest if needed
2EBC 21002D          LD      HL,BUF3$        ;Set up to read
2EBF 55              LD      D,L             ;  track 0,
2EC0 5D              LD      E,L             ;  sector 0
2EC1 CD7228          CALL    RDSEC
2EC4 C29726          JP      NZ,EXIT3        ;Quit on read error
2EC7 3676            LD      (HL),76H        ;Insert HALT to guard
2EC9 21002D          LD      HL,BUF3$        ; against incomplete BU
2ECC CD6828          CALL    WRSEC
2ECF C29726          JP      NZ,EXIT3        ;Quit on write error
2ED2 3A022D          LD      A,(BUF3$+2)     ;P/U current dest/dir
2ED5 320431          LD      (STRDIR$+1),A   ; store it for later
               ;
               ;     Use source directory track for destination
               ;
2ED8 CD1A27          CALL    PMTSRC          ;Prompt source
2EDB FD7E09          LD      A,(IY+9)        ;Get source dir cyl
2EDE 320F30          LD      (DSTDIR+1),A
               ;
               ;     Calculate the number of sectors per cylinder
               ;
2EE1 FD7E07          LD      A,(IY+7)        ;P/u # of sectors per cyl
2EE4 47              LD      B,A             ;Save # heads also
2EE5 E61F            AND     1FH             ;Mask all but sectors
2EE7 4F              LD      C,A
2EE8 0C              INC     C               ;Adj for zero offset
2EE9 A8              XOR     B               ;Get # of heads
2EEA 07              RLCA
2EEB 07              RLCA                    ;Shift to bits 0-2
2EEC 07              RLCA
2EED 3C              INC     A               ;Adj for 0 offset
2EEE 47              LD      B,A             ;Init loop counter
2EEF AF              XOR     A               ;Set sector count to 0
2EF0 81              ADD     A,C             ;Multiply # sectors/track
2EF1 10FD            DJNZ    $-1             ;X # of heads/cyl
2EF3 FDCB046E        BIT     5,(IY+4)        ;If 2-sided diskette
2EF7 2801            JR      Z,$+3
2EF9 87              ADD     A,A             ;Double the # of sectors
2EFA 327E2F          LD      (LDCYL4+1),A    ;Save sect/cyl total
2EFD 322630          LD      (DUCYL5+1),A    ;  in many places
2F00 327B30          LD      (VECYL4+1),A
2F03 329031          LD      (RESMF6+1),A
2F06 32F130          LD      (RESMF2+1),A
               ;
               ;     Calculate the amount of core available
               ;
2F09 47              LD      B,A             ;Put sector count in B
2F0A 210000          LD      HL,0            ;Set up to get HIGH$
2F0D C5              PUSH    BC              ;Save the count
2F0E 45              LD      B,L
2F0F                 @@HIGH$                 ;Get HIGH$
2F0F 3E64            LD      A,100
2F11 EF              RST     40
2F12 C1              POP     BC              ;Recover sector count
2F13 23              INC     HL              ;Get highest full page
2F14 25              DEC     H
2F15 ED5B1626        LD      DE,(BUFFER$)    ;Get buffer addr
2F19 7C              LD      A,H             ;Now sub buffer start
2F1A 92              SUB     D               ;  from the top
```

```
2F1B ØEFF                LD      C,-1
2F1D ØC          $A2     INC     C               ;Now count how many cyls
2F1E 9Ø                  SUB     B               ;  will fit in this space
2F1F 3ØFC                JR      NC,$A2
2F21 79                  LD      A,C             ;This is the number of full
2F22 32892F              LD      (LDCYL6+1),A    ;  cylinders to move per pass
                 ;
                 ;       Get source & initialize
                 ;
2F25 CD1A27              CALL    PMTSRC          ;Prompt source if needed
2F28 AF                  XOR     A               ;Init starting cylinder
2F29 32822F              LD      (LDCYL5+1),A    ;  to Ø
2F2C 57                  LD      D,A             ;Set current track to Ø
2F2D CD8928              CALL    CKSWDD
                 ;
                 ;       Here each time a new load cycle
                 ;
2F3Ø 2A1626      LDTKS   LD      HL,(BUFFER$)    ;Pt to buffer start
2F33 7A                  LD      A,D             ;P/u cylinder to move
2F34 32A52F              LD      (DUCYL+1),A     ;Save start for dump cycle
                 ;
                 ;       Here on each track loaded
                 ;
                 LDTKS1
2F37                     @@CKBRKC                ;Ckeck for break
2F37 3E6A                LD      A,1Ø6
2F39 EF                  RST     4Ø
2F3A C2AC26              JP      NZ,BREAK        ;  and abort if so
                 ;
2F3D E5                  PUSH    HL              ;Save buffer
2F3E 262B                LD      H,BUF1$<-8      ;Pt to source GAT
2F4Ø 6A                  LD      L,D             ;  for this cylinder
2F41 4E                  LD      C,(HL)          ;P/u Free/used byte
2F42 7A                  LD      A,D
2F43 C66Ø                ADD     A,6ØH           ;Pt to Lockout byte
2F45 6F                  LD      L,A             ;If source track is
2F46 7E                  LD      A,(HL)          ;  locked out, don't
2F47 2F                  CPL                     ;  back it up - BUT
2F48 A1                  AND     C               ;  show dest is "in use"
2F49 262C                LD      H,BUF2$<-8      ;Pt to dest lockout
2F4B 4E                  LD      C,(HL)          ;P/u dest lockout byte
2F4C B1                  OR      C               ;Merge with source
2F4D 6A                  LD      L,D             ;Xfer pattern to FREE
2F4E 77                  LD      (HL),A          ;  field of dest
2F4F B9                  CP      C
2F5Ø E1                  POP     HL              ;Recover buffer
2F51 CA8D2F              JP      Z,LDCYL7        ;Go if ignore this track
                 ;
                 ;       Get source disk and load
                 ;
2F54 CD1A27              CALL    PMTSRC          ;Prompt source if needed
2F57 E5                  PUSH    HL              ;Save buffer
2F58 1EØØ                LD      E,Ø             ;Start track at sector Ø
2F5A 7A                  LD      A,D             ;This is the cylinder
2F5B 21Ø132              LD      HL,CYL$         ;Message posn to hold
2F5E CD9631              CALL    CVTDEC          ;  ASCII cyl number
2F61 D5                  PUSH    DE
2F62                     @@DSPLY LDCYL$          ;"loading cylinder...
                         IFEQ    Ø1H,1
```

Mirror Image Backup

```
2F62 21C231     ØØ142           LD      HL,LDCYL$
                ØØ143           ENDIF
2F65 3EØA       ØØ144           LD      A,1Ø
2F67 EF         ØØ145           RST     4Ø
2F68            159ØØ           @@DSPLY CYL$                ;"xx...
                ØØ146           IFEQ    Ø1H,1
2F68 21Ø132     ØØ147           LD      HL,CYL$
                ØØ148           ENDIF
2F6B 3EØA       ØØ149           LD      A,1Ø
2F6D EF         ØØ15Ø           RST     4Ø
2F6E D1         1591Ø           POP     DE                  ;Now set up to
2F6F E1         1592Ø           POP     HL                  ;  read the cylinder
2F7Ø CD7228     1593Ø LDCYL2    CALL    RDSEC               ;Read a sector
2F73 28Ø5       1594Ø           JR      Z,LDCYL3            ;Go if no error
2F75 FEØ6       1595Ø           CP      6                   ;Ok if error 6 (reading DIR
2F77 C29726     1596Ø           JP      NZ,EXIT3
2F7A 24         1597Ø LDCYL3    INC     H                   ;Bump buffer and
2F7B 1C         1598Ø           INC     E                   ;  sector number
2F7C 7B         1599Ø           LD      A,E
2F7D FEØØ       16ØØØ LDCYL4    CP      Ø                   ;High sector #
2F7F 2ØEF       16Ø1Ø           JR      NZ,LDCYL2           ;Loop til cyl. finished
2F81 3EØØ       16Ø2Ø LDCYL5    LD      A,$-$               ;P/u current cylinder
2F83 3C         16Ø3Ø           INC     A
2F84 32822F     16Ø4Ø           LD      (LDCYL5+1),A        ;Store next cyl
2F87 47         16Ø5Ø           LD      B,A
2F88 3EØØ       16Ø6Ø LDCYL6    LD      A,$-$               ;P/u last for this pass
2F8A B8         16Ø7Ø           CP      B                   ;See if memory full
2F8B 28ØE       16Ø8Ø           JR      Z,LDCYL8            ;  and go if so
2F8D 14         16Ø9Ø LDCYL7    INC     D                   ;Bump cyl to use
2F8E 7A         161ØØ           LD      A,D
2F8F FE6Ø       1611Ø           CP      6ØH                 ;Highest track #?
2F91 C2372F     1612Ø           JP      NZ,LDTKS1           ;If not, do another
2F94 3A822F     1613Ø           LD      A,(LDCYL5+1)        ;Were any moved?
2F97 B7         1614Ø           OR      A                   ;Don't dump if not
2F98 CA8F3Ø     1615Ø           JP      Z,MOVID
2F9B 3A822F     1616Ø LDCYL8    LD      A,(LDCYL5+1)        ;P/u last cyl loaded
2F9E 327F3Ø     1617Ø           LD      (VECYL5+1),A        ;  & save for VERIFY
                1618Ø ;
                1619Ø ;         Get ready to dump to destination
                162ØØ ;
2FA1 2A1626     1621Ø           LD      HL,(BUFFER$)        ;P/u start of buffer
2FA4 16ØØ       1622Ø DUCYL     LD      D,$-$               ;Init starting cylinder
                1623Ø ;
                1624Ø DUCYL1
2FA6            1625Ø           @@CKBRKC                    ;Check for break
2FA6 3E6A       ØØ151           LD      A,1Ø6
2FA8 EF         ØØ152           RST     4Ø
2FA9 C2AC26     1626Ø           JP      NZ,BREAK            ; and abort if hit
                1627Ø ;
                1628Ø ;         Start by making dest GAT bytes
                1629Ø ;
2FAC E5         163ØØ           PUSH    HL                  ;Save buffer ptr
2FAD 262B       1631Ø           LD      H,BUF1$<-8          ;Pt to source GAT
2FAF 6A         1632Ø           LD      L,D                 ;  at current cylinder
2FBØ 4E         1633Ø           LD      C,(HL)              ;Get the free/used byte
2FB1 7A         1634Ø           LD      A,D
2FB2 C66Ø       1635Ø           ADD     A,6ØH               ;P/u the lockout byte
2FB4 6F         1636Ø           LD      L,A                 ;  for this cylinder
2FB5 7E         1637Ø           LD      A,(HL)
```

Mirror Image Backup

```
2FB6 2F                 CPL                     ;Merge non-locked and
2FB7 A1                 AND     C               ;  in use bits
2FB8 262C               LD      H,BUF2$<-8      ;Pt to dest GAT
2FBA 4E                 LD      C,(HL)          ;P/u its lockout byte
2FBB B1                 OR      C               ;Merge in source info
2FBC 6A                 LD      L,D             ;Store in dest free/used
2FBD 77                 LD      (HL),A
2FBE B9                 CP      C               ;Check if any in use
2FBF E1                 POP     HL
2FCØ CA3Ø3Ø             JP      Z,DUCYL6        ;  and go if not
2FC3 CD8727             CALL    PMTDST          ;Set up to write dest disk
2FC6 1EØØ               LD      E,Ø             ;Init to sector Ø
2FC8 7A                 LD      A,D             ;Get current cylinder
2FC9 B3                 OR      E
2FCA E5                 PUSH    HL              ;Save buffer ptr
2FCB 7A                 LD      A,D
2FCC 21Ø132             LD      HL,CYL$         ;"xx...
2FCF CD9631             CALL    CVTDEC          ;Convert cyl # to ASCII
2FD2 D5                 PUSH    DE
2FD3                    @@DSPLY DUCYL$          ;"dumping cyl...
                        IFEQ    Ø1H,1
2FD3 21D731             LD      HL,DUCYL$
                        ENDIF
2FD6 3EØA               LD      A,1Ø
2FD8 EF                 RST     4Ø
2FD9                    @@DSPLY CYL$            ;"xx...
                        IFEQ    Ø1H,1
2FD9 21Ø132             LD      HL,CYL$
                        ENDIF
2FDC 3EØA               LD      A,1Ø
2FDE EF                 RST     4Ø
2FDF D1                 POP     DE              ;Recover cyl/sect
2FEØ E1                 POP     HL              ;  and buffer posn
2FE1 7A         DUCYL2  LD      A,D             ;P/u track # & bypass
2FE2 B7                 OR      A               ;  if not cyl=Ø
2FE3 2Ø28               JR      NZ,DUCYL2B
                ;
                        IF      @MOD2
                        LD      A,(BACKUPØ)     ;Get system flag
                        OR      A               ;System disk?
                        JR      NZ,DUCYL2B      ;Yes, bypass!
                        ENDIF
                ;
2FE5 B3                 OR      E               ;Merge to test for sec=2
2FE6 FEØ2               CP      2
2FE8 2ØØD               JR      NZ,CKBOOT       ;If not 2, ck 1 or Ø
2FEA 2EC6               LD      L,ØC6H          ;Point to id byte
2FEC 7E                 LD      A,(HL)
2FED 3C                 INC     A               ;If X'FF', leave as is
2FEE 2818               JR      Z,SETØ
2FFØ 3D                 DEC     A               ;If X'ØØ', leave as is
2FF1 2815               JR      Z,SETØ
2FF3 36FF               LD      (HL),-1         ;Set to X'FF'
2FF5 1811               JR      SETØ
2FF7 E6FE       CKBOOT  AND     ØFEH            ;Sector Ø or 1?
2FF9 2Ø12               JR      NZ,DUCYL2B      ;Go if not
2FFB B3                 OR      E               ;If sector Ø, just
2FFC 28ØD               JR      Z,DUCYL2A       ;  bother with HALT
                ;
```

Mirror Image Backup

```
                16870 ;         Keep the boot track step rate
                16880 ;
2FFE 3AØØ26     1689Ø           LD      A,(BOOTST$)       ;P/u step pointer
3ØØ1 6F         169ØØ           LD      L,A               ;  & update buffer ptr
3ØØ2 7E         1691Ø           LD      A,(HL)            ;P/u this step byte
3ØØ3 E6FC       1692Ø           AND     ØFCH              ;  & strip the step rate
3ØØ5 F6ØØ       1693Ø BSMIR     OR      Ø                 ;Merge with the step
3ØØ7 77         1694Ø           LD      (HL),A
3ØØ8 2EØØ       1695Ø SETØ      LD      L,Ø               ;Reset buffer pointer
3ØØA Ø1         1696Ø           DB      1                 ;Ignore next via LD BC,nn
3ØØB 3676       1697Ø DUCYL2A   LD      (HL),76H          ;Keep the HALT in dest
3ØØD 7A         1698Ø DUCYL2B   LD      A,D               ;P/u the cylinder #
3ØØE FEØØ       1699Ø DSTDIR    CP      Ø                 ;Is this the dir cyl?
3Ø1Ø 28Ø8       17ØØØ           JR      Z,DUCYL3          ;Go if it is
3Ø12 CD6828     17Ø1Ø           CALL    WRSEC             ;Write non-dir sector
3Ø15 C29726     17Ø2Ø           JP      NZ,EXIT3          ;Quit on write error
3Ø18 18Ø8       17Ø3Ø           JR      DUCYL4
3Ø1A CD6D28     17Ø4Ø DUCYL3    CALL    WRSYS             ;Write dir sector
3Ø1D 3E12       17Ø5Ø           LD      A,18              ;Init "Dir write error
3Ø1F C29726     17Ø6Ø           JP      NZ,EXIT3          ;  and leave if error
3Ø22 24         17Ø7Ø DUCYL4    INC     H                 ;Advance buffer and
3Ø23 1C         17Ø8Ø           INC     E                 ;  sector #
3Ø24 7B         17Ø9Ø           LD      A,E
3Ø25 FEØØ       171ØØ DUCYL5    CP      Ø                 ;Reach end of cylinder?
3Ø27 2ØB8       1711Ø           JR      NZ,DUCYL2         ;Go if not
3Ø29 3A822F     1712Ø           LD      A,(LDCYL5+1)      ;Count down one more
3Ø2C 3D         1713Ø           DEC     A                 ;  cylinder dumped
3Ø2D 32822F     1714Ø           LD      (LDCYL5+1),A
3Ø3Ø 14         1715Ø DUCYL6    INC     D                 ;Bump cylinder #
3Ø31 3A822F     1716Ø           LD      A,(LDCYL5+1)      ;Loop if still more
3Ø34 B7         1717Ø           OR      A                 ; to dump
3Ø35 C2A62F     1718Ø           JP      NZ,DUCYL1
                1719Ø ;
                172ØØ ;         Prepare to verify
                1721Ø ;
3Ø38 3AA52F     1722Ø           LD      A,(DUCYL+1)       ;P/u cyl # to start
3Ø3B 57         1723Ø           LD      D,A
                1724Ø VECYL1
3Ø3C            1725Ø           @@CKBRKC                  ;Check if Break hit
3Ø3C 3E6A       ØØ163           LD      A,1Ø6
3Ø3E EF         ØØ164           RST     4Ø
3Ø3F C2AC26     1726Ø           JP      NZ,BREAK          ;Abort on break
                1727Ø ;
3Ø42 262B       1728Ø           LD      H,BUF1$<-8        ;Pt to source GAT
3Ø44 6A         1729Ø           LD      L,D               ;  at the current cylinder
3Ø45 4E         173ØØ           LD      C,(HL)            ;Get free/used byte
3Ø46 7A         1731Ø           LD      A,D
3Ø47 C66Ø       1732Ø           ADD     A,6ØH             ;Pt to lockout byte for
3Ø49 6F         1733Ø           LD      L,A               ;  the current cylinder
3Ø4A 7E         1734Ø           LD      A,(HL)            ;P/u the locked out info
3Ø4B 2F         1735Ø           CPL                       ;Merge the non-locked and
3Ø4C A1         1736Ø           AND     C                 ;  and the free/ used
3Ø4D 262C       1737Ø           LD      H,BUF2$<-8        ;Pt to dest GAT
3Ø4F 4E         1738Ø           LD      C,(HL)            ;P/u lockout for dest cyl
3Ø5Ø B1         1739Ø           OR      C                 ;Merge source info
3Ø51 6A         174ØØ           LD      L,D               ;Pt to dest free/used
3Ø52 77         1741Ø           LD      (HL),A            ;  and store new value
3Ø53 B9         1742Ø           CP      C                 ;See if in use
3Ø54 CA843Ø     1743Ø           JP      Z,VECYL6          ;Skip verify if not
```

Mirror Image Backup

```
3Ø57 1EØØ       1744Ø          LD       E,Ø                ;Init to sector Ø
3Ø59 7A         1745Ø          LD       A,D                ;P/u cyl # for dsply
3Ø5A 21Ø132     1746Ø          LD       HL,CYL$            ;"xx...
3Ø5D CD9631     1747Ø          CALL     CVTDEC             ;Convert cyl # to ASCII
3Ø6Ø D5         1748Ø          PUSH     DE
3Ø61            1749Ø          @@DSPLY  VECYL$             ;"verifying cyl...
                ØØ165          IFEQ     Ø1H,1
3Ø61 21EC31     ØØ166          LD       HL,VECYL$
                ØØ167          ENDIF
3Ø64 3EØA       ØØ168          LD       A,1Ø
3Ø66 EF         ØØ169          RST      4Ø
3Ø67            175ØØ          @@DSPLY  CYL$               ;"xx...
                ØØ17Ø          IFEQ     Ø1H,1
3Ø67 21Ø132     ØØ171          LD       HL,CYL$
                ØØ172          ENDIF
3Ø6A 3EØA       ØØ173          LD       A,1Ø
3Ø6C EF         ØØ174          RST      4Ø
3Ø6D D1         1751Ø          POP      DE                 ;Recover cyl/sector
3Ø6E CD7728     1752Ø VECYL2   CALL     VERSEC             ;Verify a sector
3Ø71 28Ø5       1753Ø          JR       Z,VECYL3           ;Go if no error
3Ø73 FEØ6       1754Ø          CP       6                  ;Error 6 is OK
3Ø75 C29726     1755Ø          JP       NZ,EXIT3
3Ø78 1C         1756Ø VECYL3   INC      E                  ;Inc sector #
3Ø79 7B         1757Ø          LD       A,E
3Ø7A FEØØ       1758Ø VECYL4   CP       Ø                  ;Check end of cylinder
3Ø7C 2ØFØ       1759Ø          JR       NZ,VECYL2          ;Loop if not
3Ø7E 3EØØ       176ØØ VECYL5   LD       A,Ø                ;Count down another
3Ø8Ø 3D         1761Ø          DEC      A                  ;  cyl just verified
3Ø81 327F3Ø     1762Ø          LD       (VECYL5+1),A
3Ø84 14         1763Ø VECYL6   INC      D                  ;Bump cyl # by 1
3Ø85 3A7F3Ø     1764Ø          LD       A,(VECYL5+1)       ;Loop if more cylinders
3Ø88 B7         1765Ø          OR       A                  ;  to verify, else go
3Ø89 C23C3Ø     1766Ø          JP       NZ,VECYL1          ;  back to "loading"
3Ø8C C33Ø2F     1767Ø          JP       LDTKS
                1768Ø ;
                1769Ø ;        All cylinders backed up, move ID info
                177ØØ ;
3Ø8F ØEØD       1771Ø MOVID    LD       C,CR               ;Print a newline
3Ø91            1772Ø          @@DSP
3Ø91 3EØ2       ØØ175          LD       A,2
3Ø93 EF         ØØ176          RST      4Ø
3Ø94 21CD2B     1773Ø          LD       HL,BUF1$+ØCDH      ;Move in the pswd,name,
3Ø97 11CD2C     1774Ø          LD       DE,BUF2$+ØCDH      ;  date, "AUTO" buffer,
3Ø9A Ø133ØØ     1775Ø          LD       BC,33H             ;  & config byte
3Ø9D EDBØ       1776Ø          LDIR
3Ø9F 217826     1777Ø          LD       HL,DATFLD$         ;Move in today's date
3ØA2 11D82C     1778Ø          LD       DE,BUF2$+ØD8H
3ØA5 ØEØ8       1779Ø          LD       C,8
3ØA7 EDBØ       178ØØ          LDIR
                1781Ø ;
                1782Ø ;        Get destination disk & write new GAT
                1783Ø ;
3ØA9 CD8727     1784Ø          CALL     PMTDST             ;Set up to use dest disk
3ØAC 3AØF3Ø     1785Ø          LD       A,(DSTDIR+1)       ;Get dir cyl
3ØAF 57         1786Ø          LD       D,A                ;Set to track Dir,
3ØBØ 1EØØ       1787Ø          LD       E,Ø                ;  sector Ø
3ØB2 21ØØ2C     1788Ø          LD       HL,BUF2$           ;Write the GAT back
3ØB5 CD6D28     1789Ø          CALL     WRSYS
3ØB8 3E15       179ØØ          LD       A,21               ;Init "GAT write error
```

Mirror Image Backup

```
3ØBA C29726              JP      NZ,EXIT3         ;  and go if bad
3ØBD 21ØØ2D              LD      HL,BUF3$
3ØCØ CD7728              CALL    VERSEC           ;  else verify gat
3ØC3 FEØ6                CP      6                ;Expect error 6
3ØC5 3E14                LD      A,2Ø             ;Init "GAT read error now
3ØC7 C29726              JP      NZ,EXIT3         ;  and quit if bad verify
3ØCA 3AØF3Ø              LD      A,(DSTDIR+1)     ;P/u cyl to use for dir
3ØCD 57                  LD      D,A              ;Set track = Dir
3ØCE 1EØ2                LD      E,2              ;Skip GAT and HIT
                 ;
                 ;       Reset all mod flags on destination
                 ;
3ØDØ 2A1626      RESMF   LD      HL,(BUFFER$)     ;Use this for sector buffer
3ØD3 CD7228              CALL    RDSEC            ;Read in dir record
3ØD6 FEØ6                CP      6                ;Expect error 6
3ØD8 C29226              JP      NZ,DIRERR        ;Abort on any other
3ØDB 2C                  INC     L                ;DIR+1 holds mod flag
3ØDC CBB6        RESMF1  RES     6,(HL)           ;Reset the flag
3ØDE 7D                  LD      A,L
3ØDF C62Ø                ADD     A,2ØH            ;Index to next direc
3ØE1 6F                  LD      L,A
3ØE2 3ØF8                JR      NC,RESMF1        ;  and loop thru all 8
3ØE4 2EØØ                LD      L,Ø
3ØE6 CD6D28              CALL    WRSYS            ;Write record back out
3ØE9 3E12                LD      A,18             ;Init "DIR write error
3ØEB C29726              JP      NZ,EXIT3
3ØEE 1C                  INC     E                ;Inc dir sector #
3ØEF 7B                  LD      A,E
3ØFØ FEØØ        RESMF2  CP      $-$              ;Compare highest sect this cyl
3ØF2 2ØDC                JR      NZ,RESMF         ;Loop until complete
                 ;
                         IF      @MOD2
                         LD      A,(STRDIR$+1)    ;Get old dir cyl
                         LD      B,A              ;Pass for jump
                         LD      A,(BACKUPØ)      ;Get system backup flag
                         OR      A                ;System disk?
                         JR      NZ,CNTBAK1       ;Yes, check if dir change
                         ENDIF
                 ;
                 ;       Clear the HALT inst from dest
                 ;
3ØF4 21ØØ2D              LD      HL,BUF3$
3ØF7 55                  LD      D,L              ;Now read the BOOT
3ØF8 5D                  LD      E,L              ;  on the dest disk
3ØF9 CD7228              CALL    RDSEC
3ØFC C29726              JP      NZ,EXIT3         ;Quit if couldn't be read
3ØFF 36ØØ                LD      (HL),Ø           ;Clear the HALT
31Ø1 23                  INC     HL
31Ø2 23                  INC     HL               ;Pt to old DIR cyl
31Ø3 Ø6ØØ        STRDIR$ LD      B,$-$            ;P/u the old DIR cyl
31Ø5 3AØF3Ø              LD      A,(DSTDIR+1)     ;Update the dir cyl
31Ø8 77                  LD      (HL),A           ;  in case it changed
31Ø9 2B                  DEC     HL               ;Pt back to buffer start
31ØA 2B                  DEC     HL
31ØB CD6828              CALL    WRSEC            ;Write back the BOOT
31ØE CC7728              CALL    Z,VERSEC         ;  and then verify it
3111 C29726              JP      NZ,EXIT3         ;Go if write error
3114 1C                  INC     E                ;Point to sector 1
3115 CD7228              CALL    RDSEC            ;Read it
```

Mirror Image Backup

```
3118 C29726              JP      NZ,EXIT3
311B 3AØF3Ø              LD      A,(DSTDIR+1)    ;Do the same thing again
311E 23                  INC     HL
311F 23                  INC     HL
312Ø 77                  LD      (HL),A          ;Store new dir cyl
3121 2B                  DEC     HL
3122 2B                  DEC     HL
3123 CD6828              CALL    WRSEC           ;Write it back
3126 CC7728              CALL    Z,VERSEC        ;Verify it if written OK
3129 C29726              JP      NZ,EXIT3        ;Quit if we couldn't
                 ;
312C 262C        CNTBAK1 LD      H,BUF2$<-8      ;Destination GAT
312E 78                  LD      A,B             ;P/u old DIR cyl
312F C66Ø                ADD     A,6ØH           ;Point to lockout table
3131 6F                  LD      L,A
3132 4E                  LD      C,(HL)          ;Check lockout byte
3133 68                  LD      L,B             ;Pt to GAT byte
3134 7E                  LD      A,(HL)          ;Get GAT byte
3135 B1                  OR      C
3136 B9                  CP      C               ;Anything allocated?
3137 2Ø1E                JR      NZ,RESMF2B      ;Bypass if yes
3139 78                  LD      A,B             ;Save cylinder
313A 21ØØ2D              LD      HL,BUF3$        ;Write E5's to cylinder
313D 11Ø12D              LD      DE,BUF3$+1      ; to remove system DAM
314Ø Ø1FFØØ              LD      BC,255
3143 36E5                LD      (HL),ØE5H
3145 EDBØ                LDIR
3147 69                  LD      L,C             ;Pt back to buf3$
3148 57                  LD      D,A             ;Set cylinder # in D
3149 59                  LD      E,C             ;Start with sector Ø
314A 3A7E2F              LD      A,(LDCYL4+1)    ;Get # of sectors
314D 47                  LD      B,A             ;Set loop counter
314E CD6828      RESMF2A CALL    WRSEC           ;Write normal sector
3151 C29726              JP      NZ,EXIT3
3154 1C                  INC     E               ;Step to next sector
3155 1ØF7                DJNZ    RESMF2A
3157 CD5E28      RESMF2B CALL    RESTOR          ;Restore to track Ø
                 ;
                 ;       Attempt to clear MOD flags of source
                 ;
315A CD1A27              CALL    PMTSRC          ;Set up for source disk
315D FD56Ø9              LD      D,(IY+9)        ;Get track = Dir
                 ;
316Ø 1EØ2                LD      E,2             ;Skip GAT and HIT
3162 2A1626      RESMF3  LD      HL,(BUFFER$)    ;Use this as sector buffer
3165 CD7228              CALL    RDSEC           ;Read source dir sector
3168 FEØ6                CP      6               ;Expect error 6
316A C29226              JP      NZ,DIRERR
316D 2C                  INC     L               ;Pt to DIR + 1
316E CBB6        RESMF4  RES     6,(HL)          ;Turn off mod flag
317Ø 7D                  LD      A,L
3171 C62Ø                ADD     A,2ØH           ;Index to next direc
3173 6F                  LD      L,A
3174 3ØF8                JR      NC,RESMF4       ;Loop 8 times/sector
3176 2EØØ                LD      L,Ø
3178 CD6D28              CALL    WRSYS           ;Write back dir sector
317B 281Ø                JR      Z,RESMF5        ;Loop on no error
317D FEØF                CP      15              ;Write protected source?
317F 3E12                LD      A,18            ;Init "DIR write error"
```

Mirror Image Backup

```
3181 C29726    19Ø9Ø          JP      NZ,EXIT3        ;Exit if not WP error
3184           191ØØ          @@LOGOT CCMOD$          ;"Can't clear mod flags
               ØØ177          IFEQ    Ø1H,1
3184 21C629    ØØ178          LD      HL,CCMOD$
               ØØ179          ENDIF
3187 3EØC      ØØ18Ø          LD      A,12
3189 EF        ØØ181          RST     4Ø
               1911Ø          IF      @MOD4
318A C38526    1912Ø          JP      EXIT1           ;Backup is complete
               1913Ø          ENDIF
               1914Ø          IF      @MOD2
               1915Ø          JR      CKWRTKØ         ;Check if write cyl Ø
               1916Ø          ENDIF
318D 1C        1917Ø RESMF5   INC     E               ;Bump sector #
318E 7B        1918Ø          LD      A,E
318F FEØØ      1919Ø RESMF6   CP      $-$             ;Compare highest sect this cyl
3191 2ØCF      192ØØ          JR      NZ,RESMF3       ;Do another sector if not
               1921Ø          IF      @MOD4
3193 C38526    1922Ø          JP      EXIT1           ;Backup is complete
               1923Ø          ENDIF
               1924Ø          IF      @MOD2
               1925Ø CKWRTKØ  LD      A,(BACKUPØ)     ;Get flag
               1926Ø          OR      A               ;Anything?
               1927Ø          JP      Z,EXIT1         ;Nope, go!
               1928Ø          CALL    PMTSRC          ;Prompt for source
               1929Ø          CALL    READØ           ;Read cyl Ø
               193ØØ          JP      NZ,EXIT3        ;Go on error
               1931Ø          CALL    PMTDST          ;Prompt for dest drive
               1932Ø          CALL    FORMATØ         ;Format cylinder Ø
               1933Ø          JP      NZ,EXIT3        ;Go on disk error
               1934Ø ;
               1935Ø ;        Pass original step rate to new disk
               1936Ø ;
               1937Ø          LD      HL,(BUFFER$)    ;Get I/O buffer
               1938Ø          INC     HL              ;Bump to step rate
               1939Ø          INC     HL
               194ØØ          INC     HL              ;+3
               1941Ø          LD      A,(BSMIR+1)     ;Get step
               1942Ø          LD      (HL),A          ;Pass to buffer
               1943Ø          LD      BC,8ØH          ;Offset to sector 1
               1944Ø          ADD     HL,BC           ;Point to it
               1945Ø          LD      (HL),A          ;Pass to buffer
               1946Ø          CALL    PMTDST          ;Re-fetch DCT
               1947Ø          CALL    WRITEØ          ;Write cylinder Ø
               1948Ø          JP      NZ,EXIT3        ;Go on disk error
               1949Ø          CALL    PMTDST          ;Fetch DCT
               195ØØ          LD      A,(DSTDIR+1)    ;Get new dir cyl
               1951Ø          LD      (IY+9),A        ;Update DCT
               1952Ø          CALL    UPGATØ          ;Update GAT table
               1953Ø          JP      NZ,EXIT3        ;Go on disk error
               1954Ø          JP      EXIT1           ; else program completed
               1955Ø          ENDIF
               1956Ø ;
               1957Ø ;        Routine to convert cylinder # & message stuff
               1958Ø ;
3196 362Ø      1959Ø CVTDEC   LD      (HL),' '        ;Init to leading blank
3198 Ø664      196ØØ          LD      B,1ØØ
319A CDA831    1961Ø          CALL    CVD1
319D 362Ø      1962Ø          LD      (HL),' '        ;Init to blank
```

Mirror Image Backup

```
319F Ø6ØA      1963Ø           LD      B,1Ø
31A1 CDA831    1964Ø           CALL    CVD1
31A4 363Ø      1965Ø           LD      (HL),'Ø'        ;Init to leading Ø
31A6 Ø6Ø1      1966Ø           LD      B,1
31A8 ØEØØ      1967Ø CVD1      LD      C,Ø             ;Init digit counter
31AA 9Ø        1968Ø CVD2      SUB     B               ;Sub 1Ø's power until carry
31AB 38Ø3      1969Ø           JR      C,CVD3
31AD ØC        197ØØ           INC     C               ;  and bump count
31AE 18FA      1971Ø           JR      CVD2
31BØ 8Ø        1972Ø CVD3      ADD     A,B             ;Add back last sub
31B1 F5        1973Ø           PUSH    AF
31B2 79        1974Ø           LD      A,C             ;Check the count
31B3 B7        1975Ø           OR      A
31B4 28Ø3      1976Ø           JR      Z,CVD7          ;Ignore if Ø
31B6 C63Ø      1977Ø           ADD     A,3ØH           ;  else change to ASCII digit
31B8 77        1978Ø           LD      (HL),A
31B9 F1        1979Ø CVD7      POP     AF
31BA 23        198ØØ           INC     HL
31BB C9        1981Ø           RET
               1982Ø ;
               1983Ø ;         Message area
               1984Ø ;
31BC 21Ø532    1985Ø NOTMIR    LD      HL,NOTMIR$
31BF C3AF26    1986Ø           JP      EXIT4
31C2 1D        1987Ø LDCYL$    DB      29,'Reading < cylinder ',3
     52 65 61 64 69 6E 67 2Ø
     3C 2Ø 63 79 6C 69 6E 64
     65 72 2Ø Ø3
31D7 1D        1988Ø DUCYL$    DB      29,'Writing > cylinder ',3
     57 72 69 74 69 6E 67 2Ø
     3E 2Ø 63 79 6C 69 6E 64
     65 72 2Ø Ø3
31EC 1D        1989Ø VECYL$    DB      29,'Verifying cylinder ',3
     56 65 72 69 66 79 69 6E
     67 2Ø 63 79 6C 69 6E 64
     65 72 2Ø Ø3
32Ø1 3Ø        199ØØ CYL$      DB      'ØØØ',3
     3Ø 3Ø Ø3
32Ø5 ØA        1991Ø NOTMIR$ DB        LF,'Backup aborted, '
     42 61 63 6B 75 7Ø 2Ø 61
     62 6F 72 74 65 64 2C 2Ø
3216 64        1992Ø           DB      'destination not mirror-image',CR
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 6E 6F 74 2Ø 6D
     69 72 72 6F 72 2D 69 6D
     61 67 65 ØD
3233 44        1993Ø DIFID$    DB      'Destination disk ID is different: '
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 64 69 73 6B 2Ø
     49 44 2Ø 69 73 2Ø 64 69
     66 66 65 72 65 6E 74 3A
     2Ø
3255 4E        1994Ø PACKID$ DB        'Name=XXXXXXXX  Date=mm/dd/yy',CR
     61 6D 65 3D 58 58 58 58
     58 58 58 58 2Ø 2Ø 44 61
     74 65 3D 6D 6D 2F 64 64
     2F 79 79 ØD
3272 2Ø        1995Ø OLDMPW$ DB        '  Enter its Master Password'
     2Ø 45 6E 74 65 72 2Ø 69
```

Mirror Image Backup

```
     74 73 2Ø 4D 61 73 74 65
     72 2Ø 5Ø 61 73 73 77 6F
     72 64
328D 2Ø         1996Ø          DB      ' or <BREAK> to abort: ',3
     6F 72 2Ø 3C 42 52 45 41
     4B 3E 2Ø 74 6F 2Ø 61 62
     6F 72 74 3A 2Ø Ø3
32A4 41         1997Ø PMTYN$  DB      'Are you sure you want to backup to it '
     72 65 2Ø 79 6F 75 2Ø 73
     75 72 65 2Ø 79 6F 75 2Ø
     77 61 6E 74 2Ø 74 6F 2Ø
     62 61 63 6B 75 7Ø 2Ø 74
     6F 2Ø 69 74 2Ø
32CA 3C         1998Ø          DB      '<Y,N> ? ',3
     59 2C 4E 3E 2Ø 3F 2Ø Ø3
                1999Ø ;
32D3 ØØ         2ØØØØ          DC      64,Ø            ;PATCH space
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
                2ØØ1Ø ;
Ø6ØØ            2ØØ2Ø MIRSIZ  EQU     $<-8+1<8-BACKUP
                2ØØ3Ø ;
                2ØØ4Ø ;
                2ØØ5Ø ;       Adjust PC & load address for CLASS
                2ØØ6Ø ;
2EØØ            2ØØ7Ø         ORG     BACKUP
                2ØØ8Ø ;
38ØØ            2ØØ9Ø CLSBU   EQU     CORE$+MIRSIZ
2EØØ            2Ø1ØØ         LORG    CLSBU
                2Ø11Ø ;
2EØØ            2Ø12Ø         SUBTTL  '<Backup By Class>'
```

```
                 ;
2EØØ             *GET    BACKUP3:3
                 ;BACKUP3/ASM - Backup By Class
                 ;
                 ;       Find highest available memory page
                 ;
2EØØ 21ØØØØ              LD      HL,Ø              ;Set up to get HIGH$
2EØ3 45                  LD      B,L
2EØ4                     @@HIGH$
2EØ4 3E64                LD      A,1ØØ
2EØ6 EF                  RST     4Ø
2EØ7 23                  INC     HL                ;Find highest available
2EØ8 25                  DEC     H                 ;  memory page
2EØ9 7C                  LD      A,H
2EØA 32CE32              LD      (DOFILØ6+1),A     ;Save for later testing
2EØD 32E332              LD      (DOFILØ8+1),A
2E1Ø 32DA33              LD      (LSTBUF1+1),A
2E13 3EC9                LD      A,ØC9H
2E15 32BØ27              LD      (PMTDST1),A       ;Ignore dest disk test
2E18 CD8727              CALL    PMTDST            ;Prompt dest drive
                 ;
                 ;       Calculate mamximum free space per dest disk type
                 ;
2E1B FD7EØ7              LD      A,(IY+7)          ;P/u # heads & sect/trk
2E1E 47                  LD      B,A               ;Save heads
2E1F E61F                AND     1FH               ;Mask all but sectors
2E21 4F                  LD      C,A
2E22 ØC                  INC     C                 ;Adj for zero offset
2E23 A8                  XOR     B                 ;Get # of heads
2E24 Ø7                  RLCA
2E25 Ø7                  RLCA                      ;  in bits Ø-2
2E26 Ø7                  RLCA
2E27 3C                  INC     A                 ;Adj to Ø offset
2E28 47                  LD      B,A               ;Init loop counter
2E29 AF                  XOR     A                 ;Init sector count to Ø
2E2A 81                  ADD     A,C               ;Multiply # sectors/track
2E2B 1ØFD                DJNZ    $-1               ;  x # of heads/cyl
2E2D 6F                  LD      L,A
2E2E 26ØØ                LD      H,Ø               ;Xfer to 16-bit reg
2E3Ø FDCBØ46E            BIT     5,(IY+4)          ;If 2-sided diskette
2E34 28Ø1                JR      Z,$+3
2E36 29                  ADD     HL,HL             ;  double the # of sectors
2E37 FD4EØ6              LD      C,(IY+6)          ;P/u # cyls & adjust for
2E3A ØD                  DEC     C                 ;  BOOT & DIR
2E3B                     @@MUL16                   ;Calc total records
2E3B 3E5B                LD      A,91
2E3D EF                  RST     4Ø
2E3E 65                  LD      H,L               ;Results to HL
2E3F 6F                  LD      L,A
2E4Ø 225932              LD      (SIZSAV+1),HL     ;Save for later
                 ;
                 ;       Read the BOOT sector of dest disk
                 ;
2E43 11Ø1ØØ              LD      DE,1              ;Track Ø, sector 1
2E46 21ØØ2C              LD      HL,BUF2$          ;Disk buffer area
2E49 CD7228              CALL    RDSEC             ;Read the sector
2E4C C29726              JP      NZ,EXIT3          ;Quit on read error
2E4F 3AØØ26              LD      A,(BOOTST$)       ;Locn of boot step rate
2E52 6F                  LD      L,A
2E53 7E                  LD      A,(HL)            ;Get the step rate in
```

Backup By Class

```
2E54 E6Ø3       2Ø7ØØ           AND     3               ; bits Ø and 1
2E56 321932     2Ø71Ø           LD      (BSCLS+1),A     ;Save for later
2E59 3AØ22C     2Ø72Ø           LD      A,(BUF2$+2)     ;P/u dir cyl
2E5C FD77Ø9     2Ø73Ø           LD      (IY+9),A        ;Stuff into DCT
                2Ø74Ø ;
                2Ø75Ø ;         Check id type byte
                2Ø76Ø ;
2E5F CD8928     2Ø77Ø           CALL    CKSWDD
                2Ø78Ø ;
                2Ø79Ø ;         If a system backup, then check the GAT & HIT
                2Ø8ØØ ;
2E62 3A6Ø42     2Ø81Ø           LD      A,(PRMTBL$+SYSRSP)
2E65 B7         2Ø82Ø           OR      A               ;P/u SYS parm response
2E66 CA1A2F     2Ø83Ø           JP      Z,CLSBU5        ; and skip next if not SYS
                2Ø84Ø ;
                2Ø85Ø ;         If already a SYSTEM disk, don't check BOOT space
                2Ø86Ø ;
                2Ø87Ø           IF      @MOD2
                2Ø88Ø           CALL    PMTDST          ;Get dest data
                2Ø89Ø ;
                2Ø9ØØ           LD      A,(IY+3)        ;Get DCT data
                2Ø91Ø           AND     28H             ;Bit 5/3
                2Ø92Ø           CP      2ØH             ;8" floppy?
                2Ø93Ø           JR      NZ,SETSYS2      ;Go if not
                2Ø94Ø           LD      A,(IY+4)        ;Get data
                2Ø95Ø           AND     5ØH             ;Bit 6/4
                2Ø96Ø           CP      4ØH             ;DD not alien?
                2Ø97Ø SETSYS2 LD        D,Ø             ;Cyl Ø if not
                2Ø98Ø           JR      NZ,$+3          ;Go if system
                2Ø99Ø           INC     D               ;Sysinfo on cyl 1
                21ØØØ           ENDIF
                21Ø1Ø ;
2E69 21ØØ36     21Ø2Ø           LD      HL,HITBUF       ;Set disk buffer
2E6C 1EØ2       21Ø3Ø           LD      E,2             ; and sector 2
                21Ø4Ø ;
                21Ø5Ø ;         Mod II save sysinfo sector for later check
                21Ø6Ø ;
                21Ø7Ø           IF      @MOD2
                21Ø8Ø           LD      (CKPROT2),DE    ;Save cyl/sect
                21Ø9Ø           ENDIF
                211ØØ ;
                2111Ø           IF      @MOD4
2E6E CD7228     2112Ø           CALL    RDSEC           ;Read the sysinfo sector
2E71 C29726     2113Ø           JP      NZ,EXIT3        ;QUit on read error
2E74 3ACØ36     2114Ø           LD      A,(HITBUF+ØCØH) ;P/u & test the SYSTEM
2E77 3C         2115Ø           INC     A               ; disk byte. If already
2E78 FD56Ø9     2116Ø           LD      D,(IY+9)
2E7B CACC2E     2117Ø           JP      Z,CLSBUØ1       ; a system disk, bypass
                2118Ø           ENDIF
                2119Ø ;
                212ØØ           IF      @MOD2
                2121Ø ;
                2122Ø           LD      D,(IY+9)        ;P/u dir cyl
                2123Ø ;
                2124Ø           ENDIF
                2125Ø ;
2E7E 5D         2126Ø           LD      E,L             ;Set sector Ø, dir trk
2E7F CD7228     2127Ø           CALL    RDSEC           ;Read the GAT
2E82 FEØ6       2128Ø           CP      6               ;Expect error 6
```

Backup By Class

```
2E84 3E14              LD      A,20            ;Init "GAT read error
2E86 C29726            JP      NZ,EXIT3        ;Quit on any other error
                 ;
                       IF      @MOD4
2E89 0600              LD      B,0             ;Need no more
2E8B FDCB035E          BIT     3,(IY+3)        ;  if rigid drive
2E8F 2011              JR      NZ,SETSYS       ;NZ = rigid
                       ENDIF
                 ;
                 ;     Check GAT byte on Mod2/12
                       IF      @MOD2
                       LD      L,0CDH
                       BIT     7,(HL)
                       LD      L,0
                       JP      Z,CLSBU01       ;Go if system disk
                       ENDIF
                 ;
                 ;     If ALIEN or NOT 8" space is OK
                 ;
                       IF      @MOD2
                       LD      A,(CKPROT2+1)
                       OR      A
                       JR      Z,SETSYS        ;Go if not
                       ENDIF
                 ;
                 ;     Mod II must have track 0 fully available
                 ;
                       IF      @MOD2
                       LD      A,(HITBUF+60H)  ;Track 0 lockout data
                       OR      1               ;Boot/sys allocation
                       CP      (HL)            ;Anything here?
                       JP      NZ,NOTSYS       ;Yes, cannot use!
                       ENDIF
                 ;
                 ;     Mod II must have 16 sectors available on cyl 1
                 ;
                       IF      @MOD2
                       INC     HL              ;Point to cyl 1
                       LD      B,3             ;2 grans SD or DD
                       ENDIF
                 ;
                 ;     Check to be sure additional grans needed for boot
                 ;     are not already allocated
                 ;
                       IF      @MOD4
2E91 0602              LD      B,2             ;If 8" SDEN or DDEN, then
                       ENDIF
                 ;
2E93 FDCB036E          BIT     5,(IY+3)        ;  need gran 1
2E97 2002              JR      NZ,$+4
2E99 0606              LD      B,6             ;5" needs grans 1 & 2
2E9B 7E                LD      A,(HL)          ;P/u GAT byte for BOOT
2E9C A0                AND     B               ;  & ck for needed space
2E9D 2048              JR      NZ,NOTSYS       ;Go if no free space
2E9F 7E                LD      A,(HL)          ;Reserve the GAT space
2EA0 B0                OR      B
2EA1 77                LD      (HL),A
                 ;
                 ;     Mod II must make force locked/used cyl 0
```

Backup By Class

```
                 ;
                         IF      @MOD2
                         LD      A,-1            ;Init
                         LD      L,Ø             ;Reset to beginning
                         LD      (HL),A          ;Allocate cyl Ø
                         LD      L,6ØH           ;Lockout table
                         LD      (HL),A          ;Lockout cyl Ø
                         ENDIF
                 ;
                 ;
                 ;       Mask the config byte "data/system" disk bit
                 ;
2EA2 2ECD        SETSYS  LD      L,ØCDH          ;Point to config byte
2EA4 CBBE                RES     7,(HL)          ;  & show system disk
2EA6 CDBF33              CALL    WRGAT
                 ;
                 ;       Adjust the allocation info for BOOT/SYS
                 ;
2EA9 1EØ2        CLSBUØ  LD      E,2             ;Read the directory
2EAB CD7228              CALL    RDSEC           ;  sector containing
2EAE FEØ6                CP      6               ;  BOOT/SYS record
2EBØ 3E11                LD      A,17            ;Init "dir read error
2EB2 C29726              JP      NZ,EXIT3
2EB5 Ø4                  INC     B               ;Code to 7 3 1
2EB6 Ø4                  INC     B               ;Code to 8 4 2
2EB7 CB28                SRA     B               ;Code to 4 2 1
2EB9 CB28                SRA     B               ;Code to 2 1 Ø
                 ;
                         IF      @MOD2
                         LD      A,(CKPROT2+1)
                         OR      A
                         JR      Z,CLSBUØ1
                         ENDIF
                 ;
                 ;       Mod II must force BOOT/SYS to new cyl 1
                 ;
                         IF      @MOD2
                 CLSBUØØ LD      L,16H           ;Cylinder start
                         LD      (HL),1          ;Force cyl 1
                         ENDIF
                 ;
2EBB 2E17                LD      L,17H           ;Point to gran alloc
2EBD 7Ø                  LD      (HL),B          ;Reset alloc
2EBE 2E14                LD      L,14H           ;Point to ERN
2ECØ 361Ø                LD      (HL),16         ;Update # BOOT records
2EC2 2EØØ                LD      L,Ø
2EC4 CD6D28              CALL    WRSYS           ;Write dir sector back
2EC7 3E12                LD      A,18            ;Init "dir write error
2EC9 C29726              JP      NZ,EXIT3        ;Exit if so
                 ;
                 ;       If OLD entered No SYS file check needed
                 ;
                 CLSBUØ1
2ECC 3A1Ø26              LD      A,(OLDPRM$)     ;Check for OLD entered
2ECF B7                  OR      A
2EDØ 2Ø48                JR      NZ,CLSBU5       ;Skip SYS setup if so
                 ;
                 ;
                 ;       Now check the HIT positions for /SYS files
```

Backup By Class

```
                   ;
2ED2 CD4F34                CALL    HITRD           ;Read in destination HIT
2ED5 C29726                JP      NZ,EXIT3
2ED8 119735                LD      DE,SYSDEC       ;Pt to SYS file hash codes
2EDB EB                    EX      DE,HL           ;HIT to DE, hash tbl to HL
2EDC 0610                  LD      B,16            ;Check 16 DECs
2EDE 1A            CLSBU1  LD      A,(DE)          ;If dest spare, stuff
2EDF B7                    OR      A               ;  with source else
2EE0 2002                  JR      NZ,CLSBU2       ;  test for match
2EE2 7E                    LD      A,(HL)
2EE3 12                    LD      (DE),A
2EE4 BE            CLSBU2  CP      (HL)            ;Dest match source?
2EE5 2806                  JR      Z,CLSBU3        ;Continue if so
2EE7 213C35        NOTSYS  LD      HL,NOTSYS$      ;Init"Can't make sys disk...
2EEA C3AF26                JP      EXIT4           ;Display and quit
2EED 1C            CLSBU3  INC     E               ;Bump to next DEC
2EEE 23                    INC     HL              ;  & our table
2EEF 3E08                  LD      A,8             ;At midpoint?
2EF1 BB                    CP      E
2EF2 2002                  JR      NZ,CLSBU4       ;Skip if not
2EF4 1E20                  LD      E,20H           ;Adjust DEC row #
2EF6 10E6          CLSBU4  DJNZ    CLSBU1
2EF8 FD5609                LD      D,(IY+9)        ;Ok to backup SYSTEM
2EFB 1E01                  LD      E,1             ;Init to HIT sector
2EFD 210036                LD      HL,HITBUF
2F00 CD6D28                CALL    WRSYS           ;Write back dest HIT
2F03 3E17                  LD      A,23            ;Init "HIT write error
2F05 CC4F34                CALL    Z,HITRD         ;Verify if write OK
2F08 C29726                JP      NZ,EXIT3        ;Quit on any error
                   ;
                   ;       Set up byte 'C0' in SYSINFO sector
                   ;
                           IF      @MOD2
                           LD      DE,(CKPROT2)    ;Get sysinfo sector
                           LD      E,2             ;Force sector 2
                           ENDIF
                   ;
                           IF      @MOD4
2F0B 110200                LD      DE,02           ;P/u Mod4 SYSINFO sect
                           ENDIF
                   ;
                   ;       HL => to HITBUF at this point
                   ;
2F0E CD7228                CALL    RDSEC           ;Read the sector
2F11 2EC0                  LD      L,0C0H          ;Point to type flag
2F13 36FF                  LD      (HL),0FFH       ;Set it
2F15 2E00                  LD      L,0             ;Reset buffer
2F17 CD6828                CALL    WRSEC           ; Write it back
                   ;
                   CLSBU5
2F1A CD1A27                CALL    PMTSRC          ;Set up for source disk
2F1D CD4F34                CALL    HITRD           ;Read source HIT
2F20 C29726                JP      NZ,EXIT3
                   ;
                   ;       Start the backup of files
                   ;
2F23 210036                LD      HL,HITBUF       ;Init to start of HIT
2F26 1834                  JR      SCNH3           ;Branch to start
2F28 D2            OPENIT  DB      'R'!80H         ;R2
```

Backup By Class

```
2F29 E1        SCNHIT  POP     HL              ;Remove top stack entry
2F2A C1        SCNH1   POP     BC              ;Recover DEC posn
2F2B 2636              LD      H,HITBUF<-8     ;HIT buf hi-order
2F2D 68                LD      L,B             ;  and lo-order
2F2E 7D        SCNH2   LD      A,L             ;Get the current DEC posn
2F2F C620              ADD     A,20H           ;Advance to next file on
2F31 6F                LD      L,A             ;  this dir sector until
2F32 3028              JR      NC,SCNH3        ;  end, then go to next
2F34 2C                INC     L               ;  dir sector in the HIT
2F35 CB6D              BIT     5,L             ;Did we go off the end?
2F37 2823              JR      Z,SCNH3         ;  (ie from 1F to 20)
2F39 3E00              LD      A,0
2F3A           SETBIT  EQU     $-1
2F3B B7                OR      A
2F3C 281B              JR      Z,TOEXIT1       ;If not, all done
2F3E CD8727            CALL    PMTDST          ;Get dest DCT in IY
2F41 210036            LD      HL,HITBUF
2F44 FD5609            LD      D,(IY+9)        ;Get dir cyl
2F47 5D                LD      E,L             ;Point to GAT sector
2F48 CD7228            CALL    RDSEC           ;  & read it
2F4B FE06              CP      6
2F4D 3E14              LD      A,20            ;Init "GAT read error
2F4F C29726            JP      NZ,EXIT3
2F52 2ECD              LD      L,0CDH          ;Point to config byte
2F54 CBE6              SET     4,(HL)
2F56 CDBF33            CALL    WRGAT
2F59 C38526    TOEXIT1 JP      EXIT1
               ;
               ;       Continue to scan the major loop
               ;
2F5C 7E        SCNH3   LD      A,(HL)          ;Is HIT entry spare?
2F5D B7                OR      A
2F5E 28CE              JR      Z,SCNH2         ;Loop back if so
2F60 7D                LD      A,L
2F61 E6FE              AND     0FEH            ;Bypass if BOOT or DIR
2F63 28C9              JR      Z,SCNH2
2F65 45                LD      B,L             ;Save DEC
2F66 C5                PUSH    BC
2F67 CD1A27            CALL    PMTSRC          ;Set up for source disk
2F6A FD5609            LD      D,(IY+9)        ;P/u DIR cyl
2F6D 78                LD      A,B             ;Pt to dir sector of
2F6E E61F              AND     1FH             ;  this file
2F70 C602              ADD     A,2             ;Adj for GAT & HIT
2F72 5F                LD      E,A
2F73 21002C            LD      HL,BUF2$        ;Read dir sector
2F76 CD7228            CALL    RDSEC
2F79 FE06              CP      6               ;Proper errcod?
2F7B C29226            JP      NZ,DIRERR
2F7E 78                LD      A,B             ;Pt to dir record for
2F7F E6E0              AND     0E0H            ;  the source file
2F81 6F                LD      L,A
2F82 262C              LD      H,BUF2$<-8      ;Pt to hi-order dir buf
2F84 7E                LD      A,(HL)          ;Ignore file if not
2F85 326D31            LD      (ATTRIB+1),A    ;  assigned in directory
2F88 CB67              BIT     4,A
2F8A 2831              JR      Z,NODOIT
2F8C CB7F              BIT     7,A             ;Ignore file if FXDE
2F8E C22A2F            JP      NZ,SCNH1
2F91 2C                INC     L               ;Bump to DIR+1
```

Backup By Class

```
2F92 3A1226     2365Ø           LD      A,(MODPRM$)     ;Bypass if Mod parm
2F95 B7         2366Ø           OR      A               ;  not entered
2F96 28Ø4       2367Ø           JR      Z,SCNH4
2F98 CB76       2368Ø           BIT     6,(HL)          ;If Mod parm and bit not set
2F9A 2821       2369Ø           JR      Z,NODOIT        ;  skip the file
                237ØØ ;
2F9C CB66       2371Ø SCNH4     BIT     4,(HL)          ;Check date not current
2F9E 28Ø9       2372Ø           JR      Z,SCNH4A
2FAØ 3AA128     2373Ø           LD      A,(SVCTR)
2FA3 B7         2374Ø           OR      A               ;Was date set?
2FA4 2817       2375Ø           JR      Z,NODOIT        ;Bypass if not
2FA6 3C         2376Ø           INC     A               ;Is date current?
2FA7 2814       2377Ø           JR      Z,NODOIT        ;Bypass if not
                2378Ø ;
2FA9 2D         2379Ø SCNH4A    DEC     L               ;DIR + Ø
2FAA 3A8426     238ØØ           LD      A,(CLSFLG$)     ;P/u CLASS parm byte
2FAD CB76       2381Ø           BIT     6,(HL)          ;Bypass if not SYS file
2FAF 28Ø6       2382Ø           JR      Z,CKINV
2FB1 CB77       2383Ø           BIT     6,A             ;Ok, it is, was SYS used?
2FB3 28Ø8       2384Ø           JR      Z,NODOIT        ;Go if no SYS parm
2FB5 18Ø9       2385Ø           JR      CKNAM           ;  else back it up
2FB7 CB5E       2386Ø CKINV     BIT     3,(HL)          ;Test if file is INV
2FB9 28Ø5       2387Ø           JR      Z,CKNAM
2FBB CB5F       2388Ø           BIT     3,A             ;File is, want INV files?
2FBD CA2A2F     2389Ø NODOIT    JP      Z,SCNH1         ;Don't want invisibles
2FCØ 3AØ226     239ØØ CKNAM     LD      A,(SPCFLD$)     ;Now test filespec match
2FC3 FE2Ø       2391Ø           CP      ' '             ;If blank, don't bother
2FC5 2ØØ7       2392Ø           JR      NZ,CKNAMØ       ;  to match, take it
2FC7 3AØA26     2393Ø           LD      A,(SPCFLD$+8)   ;How about the extension?
2FCA FE2Ø       2394Ø           CP      ' '
2FCC 282C       2395Ø           JR      Z,SCNH6         ;Go if no ext either
                2396Ø ;
                2397Ø ;         Test for a filespec match
                2398Ø ;
2FCE E5         2399Ø CKNAMØ    PUSH    HL
2FCF 7D         24ØØØ           LD      A,L
2FDØ C6Ø5       24Ø1Ø           ADD     A,5             ;Pt to filename in dir
2FD2 6F         24Ø2Ø           LD      L,A
2FD3 11Ø226     24Ø3Ø           LD      DE,SPCFLD$      ;Pt to user filespec
2FD6 Ø6ØB       24Ø4Ø           LD      B,11            ;11 char max
2FD8 1A         24Ø5Ø CKNAM1    LD      A,(DE)          ;P/u user entry
2FD9 FE24       24Ø6Ø           CP      '$'             ;Wild card character?
2FDB 28Ø8       24Ø7Ø           JR      Z,CKNAM2        ;Always matches
2FDD BE         24Ø8Ø           CP      (HL)            ;Same as filespec?
2FDE 28Ø5       24Ø9Ø           JR      Z,CKNAM2        ;Loop if so
2FEØ FE2Ø       241ØØ           CP      ' '             ;Ignore any further?
2FE2 C2F22F     2411Ø           JP      NZ,TSTMFLG      ;If not blank, no match
2FE5 23         2412Ø CKNAM2    INC     HL              ;Match so far
2FE6 13         2413Ø           INC     DE
2FE7 1ØEF       2414Ø           DJNZ    CKNAM1
                2415Ø ;
                2416Ø ;         Filespec class matches, check if NOT used
                2417Ø ;
2FE9 3AØD26     2418Ø           LD      A,(MFLG$)       ;Bypass if a match but
2FEC B7         2419Ø           OR      A               ;  - exclude given
2FED C2292F     242ØØ           JP      NZ,SCNHIT       ;- was used, skip file
2FFØ 18Ø7       2421Ø           JR      SCNH5
                2422Ø ;
2FF2 3AØD26     2423Ø TSTMFLG   LD      A,(MFLG$)       ;Ignore if NG match &
```

Backup By Class

```
2FF5 B7                       OR      A                ;  no exclude given
2FF6 CA292F                   JP      Z,SCNHIT
2FF9 E1              SCNH5    POP     HL               ;Rcvr ptr to DIR+Ø
2FFA E5              SCNH6    PUSH    HL
                     ;
                     ;        Now check if date matches
                     ;
2FFB 23                       INC     HL               ;Pt to date field
2FFC CDAE33                   CALL    UNPACK           ;Alter date for cpr
2FFF 3AØ126                   LD      A,(FTFLG$)
3ØØ2 Ø7                       RLCA                     ;Tst From bit
3ØØ3 3Ø1Ø                     JR      NC,SCNH7
3ØØ5 7A                       LD      A,D              ;Ignore if date was
3ØØ6 B3                       OR      E                ;  ØØ/ØØ/ØØ for file
3ØØ7 CA292F                   JP      Z,SCNHIT
3ØØA 2A8Ø26                   LD      HL,(FMPAKD$)     ;P/u user entry
3ØØD EB                       EX      DE,HL
3ØØE CD8233                   CALL    CPHLDE           ;HL-DE
3Ø11 EB                       EX      DE,HL
3Ø12 DA292F                   JP      C,SCNHIT         ;Bypass if date range bad
3Ø15 3AØ126          SCNH7    LD      A,(FTFLG$)
3Ø18 ØF                       RRCA                     ;Test TO bit
3Ø19 3ØØE                     JR      NC,MATCHES       ;Go if no TOPARM else
3Ø1B 7A                       LD      A,D              ;  ck if file is dated
3Ø1C B3                       OR      E
3Ø1D CA292F                   JP      Z,SCNHIT         ;Bypass if date was ØØ
3Ø2Ø 2A8226                   LD      HL,(TOPAKD$)     ;P/u user's packed date
3Ø23 CD8233                   CALL    CPHLDE           ;HL-DE
3Ø26 DA292F                   JP      C,SCNHIT         ;Bypass if out of range
3Ø29 E1              MATCHES  POP     HL
3Ø2A 7D              DONAM    LD      A,L              ;Pt to start of dir rec
3Ø2B E6EØ                     AND     ØEØH
3Ø2D 6F                       LD      L,A              ;Make sure it's on stack
3Ø2E E5                       PUSH    HL
3Ø2F C6Ø5                     ADD     A,5              ;Pt to start of filename
3Ø31 6F                       LD      L,A
3Ø32 111826                   LD      DE,FCB1$         ;Move filename into fcb
3Ø35 Ø6Ø8                     LD      B,8              ;Init 8 chars for filename
3Ø37 7E              DONAM1   LD      A,(HL)           ;P/u a char from the dir
3Ø38 FE2Ø                     CP      ' '              ;Space = end of name
3Ø3A 28Ø5                     JR      Z,DONAM2
3Ø3C 12                       LD      (DE),A           ;Move char to FCB
3Ø3D 23                       INC     HL               ;Bump both ptrs
3Ø3E 13                       INC     DE
3Ø3F 1ØF6                     DJNZ    DONAM1           ;Loop for more
3Ø41 7D              DONAM2   LD      A,L              ;Pt to file extension
3Ø42 8Ø                       ADD     A,B              ;  by adding the
3Ø43 6F                       LD      L,A              ;  loop remainder
3Ø44 7E                       LD      A,(HL)
3Ø45 FE2Ø                     CP      ' '
3Ø47 281Ø                     JR      Z,DONAM5         ;Bypass if none there
3Ø49 3E2F                     LD      A,'/'            ;  else set separator
3Ø4B 12                       LD      (DE),A           ;  into the FCB
3Ø4C 13                       INC     DE
3Ø4D Ø6Ø3                     LD      B,3              ;Now move in ext
3Ø4F 7E              DONAM4   LD      A,(HL)           ;P/u ext char
3Ø5Ø FE2Ø                     CP      ' '              ;End if no more
3Ø52 28Ø5                     JR      Z,DONAM5
3Ø54 12                       LD      (DE),A           ;Put in in the FCB
```

Backup By Class

```
3Ø55 23                         INC     HL              ;Bump both ptrs
3Ø56 13                         INC     DE
3Ø57 1ØF6                       DJNZ    DONAM4          ;Loop for more
3Ø59 3EØ3               DONAM5  LD      A,3             ;Terminate with ETX
3Ø5B 12                         LD      (DE),A
3Ø5C D5                         PUSH    DE              ;Save pointer to spec end
                        ;
                        ;       Check for NEW or OLD option
                        ;
3Ø5D 3A1Ø26                     LD      A,(OLDPRM$)     ;P/u parm & merge
3Ø6Ø 21ØE26                     LD      HL,NEWPRM$      ;  with new
3Ø63 B6                         OR      (HL)            ;If neither, bypass
3Ø64 284F                       JR      Z,BYPASS
3Ø66 211826                     LD      HL,FCB1$        ;Save current spec
3Ø69 115826                     LD      DE,FCB3$
3Ø6C Ø12ØØØ                     LD      BC,32
3Ø6F EDBØ                       LDIR
3Ø71 D1                         POP     DE              ;Recover spec end
3Ø72 D5                         PUSH    DE              ;  needed to add drivespec
3Ø73 CD8833                     CALL    MAKSPC          ;Make it a file spec
3Ø76 CD4634                     CALL    GETDST          ;Bring in the dest disk
3Ø79 2A1626                     LD      HL,(BUFFER$)    ;Buffer is irrelevant
3Ø7C 113826                     LD      DE,FCB2$        ;Pt to dest spec
3Ø7F FDE5                       PUSH    IY
3Ø81                            @@FLAGS                 ;IY => flag table base
3Ø81 3E65                       LD      A,1Ø1
3Ø83 EF                         RST     4Ø
3Ø84 FDCB12C6                   SET     Ø,(IY+'S'-'A')  ;Inhibit file open bit
3Ø88 FDE1                       POP     IY
3Ø8A                            @@OPEN                  ;Attempt to open
3Ø8A 3E3B                       LD      A,59
3Ø8C EF                         RST     4Ø
3Ø8D D1                         POP     DE              ;Keep stack proper
3Ø8E 2812                       JR      Z,CKOLD         ;If file exists, ck OLD
3Ø9Ø FE19                       CP      25              ;File access denied?
3Ø92 28ØE                       JR      Z,CKOLD         ;  means it exists
3Ø94 FE18                       CP      24              ;File not found?
3Ø96 C2292F                     JP      NZ,SCNHIT       ;Ignore if not
3Ø99 3AØE26                     LD      A,(NEWPRM$)     ;Check if NEW  requested
3Ø9C B7                         OR      A
3Ø9D 2ØØA                       JR      NZ,GODOIT       ;Go if NEW & not found
3Ø9F C3292F                     JP      SCNHIT
3ØA2 3A1Ø26             CKOLD   LD      A,(OLDPRM$)     ;Was found, backup old
3ØA5 B7                         OR      A               ; files this time?
3ØA6 CA292F                     JP      Z,SCNHIT        ;Ignore if not OLD
3ØA9 D5                 GODOIT  PUSH    DE
3ØAA 215826                     LD      HL,FCB3$        ;Recover the original
3ØAD 111826                     LD      DE,FCB1$        ;  file name
3ØBØ Ø12ØØØ                     LD      BC,32
3ØB3 EDBØ                       LDIR
                        ;
                        ;       Check if prompting or not (Q parm)
                        ;
3ØB5 3A1526             BYPASS  LD      A,(QPARM$+1)    ;Query each file?
3ØB8 B7                         OR      A
3ØB9 CA4931                     JP      Z,NOPRMPT       ;Not if not entered
3ØBC                            @@DSPLY QUERY           ;"backup filespec ?
                                IFEQ    Ø1H,1
3ØBC 217534                     LD      HL,QUERY
```

Backup By Class

```
                        ENDIF
30BF 3E0A               LD      A,10
30C1 EF                 RST     40
                ;
                ;       Display file info for user decision
                ;
30C2 D1                 POP     DE              ;Rcvr ptr to file buf
30C3 E1                 POP     HL              ;Rcvr ptr to 1st dir byte
30C4 D5                 PUSH    DE
30C5 23                 INC     HL              ;Pt to MOD bit
30C6 CB76               BIT     6,(HL)          ;Test MOD flag
30C8 2808               JR      Z,SCDAT1        ;Go if not set
30CA 3E20               LD      A,' '           ;Put a space
30CC 12                 LD      (DE),A
30CD 13                 INC     DE
30CE 3E2B               LD      A,'+'
30D0 12                 LD      (DE),A          ;Display '+' if MOD
30D1 13                 INC     DE
30D2 3E20       SCDAT1  LD      A,' '           ;Write a space
30D4 12                 LD      (DE),A
30D5 13                 INC     DE
30D6 23                 INC     HL              ;Advance to date field
30D7 EB                 EX      DE,HL
30D8 367B               LD      (HL),'{'        ;Stuff left brace
30DA 23                 INC     HL
30DB EB                 EX      DE,HL
30DC 7E                 LD      A,(HL)          ;If no date, then skip
30DD B7                 OR      A
30DE 283D               JR      Z,SCDAT4        ;Ignore if no date saved
30E0 0F                 RRCA                    ;Has date, get day
30E1 0F                 RRCA
30E2 0F                 RRCA
30E3 E61F               AND     1FH
30E5 062F               LD      B,2FH           ;Convert day to decimal
30E7 04         SCDAT2  INC     B               ;  by counting # of 10's
30E8 D60A               SUB     10              ;Sub 10 from day #
30EA 30FB               JR      NC,SCDAT2
30EC C63A               ADD     A,3AH           ;Cvrt lo order to ASCII
30EE F5                 PUSH    AF              ;Save day low order
30EF 78                 LD      A,B             ;Stuff day hi order
30F0 12                 LD      (DE),A
30F1 13                 INC     DE              ;Bump
30F2 F1                 POP     AF              ;Rcvr lo order day #
30F3 12                 LD      (DE),A          ;Stuff low order
30F4 13                 INC     DE              ;Bump pointer to msg
30F5 3E2D               LD      A,'-'
30F7 12                 LD      (DE),A          ;Stuff '-'
30F8 13                 INC     DE              ;Pt t0 month field
30F9 E5                 PUSH    HL              ;Save DIR ptr
30FA F5                 PUSH    AF              ;Save separator char
30FB 2B                 DEC     HL              ;Pt to DIR+1 (month+)
30FC 7E                 LD      A,(HL)          ;P/u month etc
30FD E60F               AND     0FH             ;Strip off flags
30FF 3D                 DEC     A               ;(mon-1)*3 to index
3100 4F                 LD      C,A             ;   string conversion table
3101 07                 RLCA                    ;X2
3102 81                 ADD     A,C             ;X3
3103 4F                 LD      C,A             ;Results to BC
3104 0600               LD      B,0
```

Backup By Class

```
3106 217335   25920         LD      HL,MONTBL       ;Ptr to month names
3109 09       25930         ADD     HL,BC           ;Add offset to tbl start
310A 0E03     25940         LD      C,3
310C EDB0     25950         LDIR                    ;Move 3-char month
310E F1       25960         POP     AF
310F 12       25970         LD      (DE),A
3110 13       25980         INC     DE              ;Advance to year field
3111 3E38     25990         LD      A,'8'           ;Stuff 8 of 1980
3113 12       26000         LD      (DE),A
3114 13       26010         INC     DE              ;Bump msg ptr
3115 E1       26020         POP     HL              ;Rcvr DIR+2
3116 7E       26030         LD      A,(HL)          ;P/u year field
3117 E607     26040         AND     7               ;Remove day
3119 C630     26050         ADD     A,'0'           ;Cvrt to ASCII
311B 12       26060         LD      (DE),A          ;Stuff -> msg
311C 13       26070         INC     DE
311D 3E03     26080 SCDAT4  LD      A,3             ;Show etx for display
311F 12       26090         LD      (DE),A
3120          26100         @@DSPLY FCB1$           ;Display filename
              00195         IFEQ    01H,1
3120 211826   00196         LD      HL,FCB1$
              00197         ENDIF
3123 3E0A     00198         LD      A,10
3125 EF       00199         RST     40
3126          26110         @@DSPLY QMARK$          ;" } ? "
              00200         IFEQ    01H,1
3126 216E35   00201         LD      HL,QMARK$
              00202         ENDIF
3129 3E0A     00203         LD      A,10
312B EF       00204         RST     40
312C 2A1626   26120         LD      HL,(BUFFER$)    ;Get user response
312F 010003   26130         LD      BC,3<8          ;3 char max
3132          26140         @@KEYIN
3132 3E09     00205         LD      A,9
3134 EF       00206         RST     40
3135 DAAC26   26150         JP      C,ABRTBU        ;Quit on Break
3138 7E       26160         LD      A,(HL)          ;Get the 1st char
3139 CBAF     26170         RES     5,A             ;Strip lc if present
313B FE59     26180         CP      'Y'             ;Yes means move the file
313D 2808     26190         JR      Z,CPYMSG        ;Go if so
              26200 ;
              26210 ;       Accept 'C' for response to set QUERY=N
              26220 ;
313F D643     26230         SUB     'C'             ;Was response "C"?
3141 C2292F   26240         JP      NZ,SCNHIT       ;Don't backup if not
3144 321526   26250         LD      (QPARM$+1),A    ;Set QUERY=N
3147 E3       26260 CPYMSG  EX      (SP),HL         ;Place dummy HL below
3148 E5       26270         PUSH    HL              ;  FCB1$ ETX pointer
              26280 ;
              26290 ;       Display copying file info
              26300 ;
3149          26310 NOPRMPT @@CKBRKC                ;Ck if BREAK
3149 3E6A     00207         LD      A,106
314B EF       00208         RST     40
314C C2AC26   26320         JP      NZ,ABRTBU       ;Quit if so
314F          26330         @@LOGOT CPYFIL$         ;"copying file...
              00209         IFEQ    01H,1
314F 216534   00210         LD      HL,CPYFIL$
              00211         ENDIF
```

Backup By Class

```
3152 3EØC      ØØ212          LD       A,12
3154 EF        ØØ213          RST      4Ø
3155 E1        2634Ø          POP      HL               ;Get pointer where ETX
3156 36ØD      2635Ø          LD       (HL),CR          ;  is & replace with CR
3158 E5        2636Ø          PUSH     HL
3159           2637Ø          @@LOGOT  FCB1$            ;Display the filespec
               ØØ214          IFEQ     Ø1H,1
3159 211826    ØØ215          LD       HL,FCB1$
               ØØ216          ENDIF
315C 3EØC      ØØ217          LD       A,12
315E EF        ØØ218          RST      4Ø
315F D1        2638Ø          POP      DE               ;Rcvr ptr to CR
316Ø E1        2639Ø          POP      HL
               264ØØ ;
               2641Ø ;        Put in the drive spec
               2642Ø ;
3161 CD8833    2643Ø DOBU     CALL     MAKSPC           ;Make the filespec
3164 C1        2644Ø          POP      BC               ;Get DEC of source
3165 C5        2645Ø          PUSH     BC
3166 78        2646Ø          LD       A,B              ;Test if a SYS DEC
3167 E6D8      2647Ø          AND      ØD8H
3169 C23A32    2648Ø          JP       NZ,DOFILØ        ;Jump if not SYS
316C 3EØØ      2649Ø ATTRIB   LD       A,Ø              ;P/u attribute byte
316E CB77      265ØØ          BIT      6,A              ;Don't do if not SYS
317Ø CA3A32    2651Ø          JP       Z,DOFILØ
               2652Ø ;
               2653Ø ;        Routine to copy over SYS files
               2654Ø ;
3173 CD8727    2655Ø          CALL     PMTDST           ;Prompt dest drive
3176 FD56Ø9    2656Ø          LD       D,(IY+9)         ;P/u dir cyl of dest
3179 78        2657Ø          LD       A,B              ;Get DEC & calc sector
317A E61F      2658Ø          AND      1FH
317C C6Ø2      2659Ø          ADD      A,2              ;Adj for GAT & HIT
317E 5F        266ØØ          LD       E,A
317F 2A1626    2661Ø          LD       HL,(BUFFER$)     ;P/u buffer addr
3182 CD7228    2662Ø          CALL     RDSEC            ;Read dir sect
3185 FEØ6      2663Ø          CP       6                ;Proper errcod?
3187 C29226    2664Ø          JP       NZ,DIRERR
318A 78        2665Ø          LD       A,B              ;Pt to 1st byte of
318B E6EØ      2666Ø          AND      ØEØH             ;  dir record
318D 6F        2667Ø          LD       L,A
318E CB66      2668Ø          BIT      4,(HL)           ;Go if already assigned
319Ø 2Ø19      2669Ø          JR       NZ,DOSYS1
3192 365F      267ØØ          LD       (HL),5FH         ;Show assigned, SYS, INV
3194 23        2671Ø          INC      HL               ;  & no access
3195 36ØØ      2672Ø          LD       (HL),Ø           ;Zero out DIR+1 to DIR+4
3197 54        2673Ø          LD       D,H
3198 5D        2674Ø          LD       E,L
3199 13        2675Ø          INC      DE
319A Ø1Ø3ØØ    2676Ø          LD       BC,3
319D EDBØ      2677Ø          LDIR
319F 7D        2678Ø          LD       A,L              ;Pt HL to DIR+16
31AØ C6ØC      2679Ø          ADD      A,12
31A2 6F        268ØØ          LD       L,A
31A3 3C        2681Ø          INC      A
31A4 5F        2682Ø          LD       E,A              ;Pt DE to DIR+17
31A5 36FF      2683Ø          LD       (HL),ØFFH        ;Stuff X'FF' into extent
31A7 ØEØF      2684Ø          LD       C,15             ;  & pswd fields
31A9 EDBØ      2685Ø          LDIR
```

Backup By Class

```
31AB 7D         DOSYS1  LD      A,L             ;Pt HL to Dir+Ø
31AC E6EØ               AND     ØEØH            ;  of dest
31AE CB76               BIT     6,(HL)          ;Guard against writing
31BØ CAE72E             JP      Z,NOTSYS        ;  over a non-SYS file
31B3 C6Ø5               ADD     A,5             ;Pt to name field
31B5 6F                 LD      L,A
31B6 5F                 LD      E,A             ;Pt DE to name field of
31B7 262C               LD      H,BUF2$<-8      ;  destination
31B9 3A1726             LD      A,(BUFFER$+1)   ;P/u buffer hi-order addr
31BC 57                 LD      D,A
31BD Ø1ØDØØ             LD      BC,13           ;Move name/ext into dest
31CØ EDBØ               LDIR
31C2 FD56Ø9             LD      D,(IY+9)        ;P/u dir cyl of dest
31C5 C1                 POP     BC              ;Rcvr DEC of source
31C6 C5                 PUSH    BC
31C7 78                 LD      A,B             ;Calc dir sector for
31C8 E61F               AND     1FH             ;  source SYS module
31CA C6Ø2               ADD     A,2
31CC 5F                 LD      E,A
31CD 2A1626             LD      HL,(BUFFER$)    ;P/u buffer ptr for dest
31DØ CD6D28             CALL    WRSYS           ;Write the dir to dest
31D3 3E12               LD      A,18            ;Init "Dir write error
31D5 C29726             JP      NZ,EXIT3        ;  and quit on bad write
                ;
                ;       The HIT entries were transferred prior
                ;
31D8 C1                 POP     BC              ;Rcvr DEC of source
31D9 C5                 PUSH    BC
31DA 78                 LD      A,B             ;Test for SYSØ
31DB FEØ2               CP      2
31DD C23A32             JP      NZ,DOFILØ       ;Bypass if not SYSØ
31EØ CD1A27             CALL    PMTSRC          ;Prompt source
                        IF      @MOD4
31E3 Ø61Ø               LD      B,16            ;Init to xfer BOOT track
31E5 11ØØØØ             LD      DE,Ø            ;Init track Ø, sector Ø
                        ENDIF
                        IF      @MOD2
                        LD      DE,(PROTSEC)    ;Get sysinfo sector
                        LD      A,D
                        OR      A
                        LD      B,5
                        JR      Z,NBTSEC2
                        LD      B,16
                NBTSEC2 LD      E,Ø
                        ENDIF
                ;
31E8 2A1626             LD      HL,(BUFFER$)    ;Set disk buffer
31EB CD7228     RDBOOT  CALL    RDSEC           ;Read sector and
31EE C29726             JP      NZ,EXIT3        ;  quit on error
31F1 24                 INC     H               ;Pt to next block
31F2 1C                 INC     E               ;Point to next sector
31F3 1ØF6               DJNZ    RDBOOT          ;Continue reading boot
                ;
                ;       Turn off CONFIG on destination disk
                ;
31F5 2A1626             LD      HL,(BUFFER$)    ;Start cyl image
31F8 11Ø1Ø2             LD      DE,1ØØH*2+1     ;Offset to sector 2 +1
31FB 19                 ADD     HL,DE           ;HL => config byte
31FC 36C9               LD      (HL),ØC9H       ;Config off
```

Backup By Class

```
                27450 ;
31FE CD8727     2746Ø DOSYS2  CALL    PMTDST          ;Prompt destination
                2747Ø         IF      @MOD4
32Ø1 Ø61Ø       2748Ø         LD      B,16            ;Sector count for boot
32Ø3 11ØØØØ     2749Ø         LD      DE,Ø            ;Init track and sector Ø
                275ØØ         ENDIF
                2751Ø         IF      @MOD2
                2752Ø         LD      DE,(CKPROT2)    ;Get dest cyl number
                2753Ø         LD      A,(PROTSEC+1)
                2754Ø         LD      B,5             ;Default 5 sectors
                2755Ø         OR      A
                2756Ø         JR      Z,NBTSECS
                2757Ø         AND     D
                2758Ø         JR      Z,NBTSECS
                2759Ø         LD      B,16            ;Use 16 sectors
                276ØØ NBTSECS LD      E,Ø
                2761Ø         ENDIF
32Ø6 2A1626     2762Ø         LD      HL,(BUFFER$)    ;P/u buffer start
32Ø9 7B         2763Ø WRBOOT  LD      A,E             ;If sector Ø or 1,
32ØA FEØ2       2764Ø         CP      2               ;  correct DIRCYL &
32ØC 3Ø15       2765Ø         JR      NC,WRBOOT2      ;  BOOT step rate
32ØE B7         2766Ø         OR      A
32ØF 28ØA       2767Ø         JR      Z,WRBOOT1       ;If sec Ø only dir cyl
                2768Ø ;
3211 3AØØ26     2769Ø         LD      A,(BOOTST$)     ;P/u step pointer
3214 6F         277ØØ         LD      L,A
3215 7E         2771Ø         LD      A,(HL)          ;P/u BOOT step rate
3216 E6FC       2772Ø         AND     ØFCH            ;Strip the rate
3218 F6ØØ       2773Ø BSCLS   OR      Ø               ;Merge dest rate
321A 77         2774Ø         LD      (HL),A
321B FD7EØ9     2775Ø WRBOOT1 LD      A,(IY+9)        ;P/u DIR cyl
321E 2EØ2       2776Ø         LD      L,2
322Ø 77         2777Ø         LD      (HL),A
3221 2EØØ       2778Ø         LD      L,Ø             ;Restart to buf start
3223 CD6828     2779Ø WRBOOT2 CALL    WRSEC           ;Write dest boot sector
3226 C29726     278ØØ         JP      NZ,EXIT3        ;Quit on error
3229 24         2781Ø         INC     H               ;Bump buffer page
322A 1C         2782Ø         INC     E               ;Bump sector
322B 1ØDC       2783Ø         DJNZ    WRBOOT
                2784Ø ;
                2785Ø ;       Verify this track
                2786Ø ;
                2787Ø         IF      @MOD4
322D Ø61Ø       2788Ø         LD      B,16            ;16 sector just written
322F 11ØØØØ     2789Ø         LD      DE,Ø            ;  on track Ø
                279ØØ         ENDIF
                2791Ø         IF      @MOD2
                2792Ø         LD      A,(PROTSEC+1)
                2793Ø         LD      B,5
                2794Ø         LD      DE,(CKPROT2)
                2795Ø         OR      A
                2796Ø         JR      Z,NBTSEC1
                2797Ø         AND     D
                2798Ø         JR      Z,NBTSEC1
                2799Ø         LD      B,16
                28ØØØ NBTSEC1 LD      E,Ø
                28Ø1Ø         ENDIF
3232 CD7728     28Ø2Ø VRBOOT  CALL    VERSEC          ;Verify a boot sector
3235 C29726     28Ø3Ø         JP      NZ,EXIT3        ;Quit on an error
```

Backup By Class

```
3238 1ØF8                  DJNZ    VRBOOT
                   ;
                   ;       Mod II check if cyl Ø to be formatted on dest
                   ;
                           IF      @MOD2
                           LD      DE,(CKPROT2)    ;Get sysinfo sector
                           LD      A,(PROTSEC+1)
                           AND     D
                           JR      Z,COPYØE        ;Go if yes
                   OKWRTØ  CALL    PMTSRC          ;Get source disk
                           CALL    READØ           ;Read cyl Ø
                           JP      NZ,EXIT3        ;Go on disk error
                           CALL    PMTDST          ;Get dest disk
                           CALL    FORMATØ         ;Format cyl
                           JP      NZ,EXIT3        ;Go on disk error
                   ;
                   ;       Setup new track length into boot data
                   ;
                           LD      HL,(BUFFER$)    ;Get I/O buffer
                           PUSH    HL              ;Save start
                           INC     HL              ;+1
                           INC     HL              ;+2 (dir cyl)
                           LD      A,(IY+9)        ;Get dir cyl
                           LD      (HL),A          ;To buffer
                           INC     HL              ;+3 (boot step rate)
                           LD      A,(BSCLS+1)     ;Get step rate
                           AND     3               ;Step rate only
                           LD      (HL),A          ;Load into buffer
                           INC     HL              ;Bump
                           LD      A,(IY+7)        ;Get data
                           AND     1FH             ;Highest sector #
                           INC     A               ;Sectors / track
                           LD      (HL),A          ;To buffer
                           INC     HL              ;Bump
                           LD      A,(IY+3)        ;Get data
                           ADD     A,A             ;Density => bit 7
                           AND     8ØH             ;Keep only
                           LD      (HL),A          ;To buffer
                           POP     HL              ;HL => buffer start
                           LD      D,H             ;Pass to DE
                           LD      E,L             ;DE => buffer start
                           LD      BC,8ØH          ;Buffer length
                           ADD     HL,BC           ;HL => dest
                           EX      DE,HL           ;HL=>source, DE=>dest
                           LDIR                    ;Copy sector Ø => sec 1
                           CALL    PMTDST          ;Re-fetch DCT
                           CALL    WRITEØ          ;Write the cylinder
                           JP      NZ,EXIT3        ;Go on disk error
                   COPYØE  EQU     $
                           ENDIF
                   ;
                   ;       Routine to perform the file copy to destination
                   ;
323A 11282F        DOFILØ  LD      DE,OPENIT       ;Check the name
323D                       @@RENAM
323D 3E38                  LD      A,56
323F EF                    RST     4Ø
324Ø Ø6ØØ                  LD      B,Ø             ;Lrl = 256
3242 CD3D34                CALL    GETSRC          ;Prompt source & set fcb
```

Backup By Class

```
3245 2A1626          LD      HL,(BUFFER$)    ;Get buffer addr
3248                 @@FLAGS
3248 3E65            LD      A,1Ø1
324A EF              RST     4Ø
324B FDCB12C6        SET     Ø,(IY+'S'-'A')  ;Inhibit file open bit
324F                 @@OPEN                  ;Open the source file
324F 3E3B            LD      A,59
3251 EF              RST     4Ø
3252 C29726          JP      NZ,EXIT3        ;Quit on open error
               ;
               ;     Check if source file can fit on destination disk
               ;
3255 2A2426          LD      HL,(FCB1$+12)   ;P/u ERN
3258 11ØØØØ   SIZSAV LD      DE,$-$          ;P/u disk capacity
325B AF              XOR     A
325C ED52            SBC     HL,DE           ;If < size, then OK
325E 38Ø9            JR      C,SIZOK
326Ø 21FC34          LD      HL,SIZBIG$      ;  else file to big
3263                 @@LOGOT                 ;Inform user & continue
                     IFEQ    ØØH,1
                     LD      HL,
                     ENDIF
3263 3EØC            LD      A,12
3265 EF              RST     4Ø
3266 C32A2F          JP      SCNH1           ;Loop back for another file
3269 11282F   SIZOK  LD      DE,OPENIT       ;Check the name
326C                 @@RENAM
326C 3E38            LD      A,56
326E EF              RST     4Ø
326F Ø6ØØ            LD      B,Ø             ;Lrl = 256
3271 CD4634          CALL    GETDST          ;Prompt dest & set fcb
3274 2A1626          LD      HL,(BUFFER$)    ;Get buffer addr
3277                 @@INIT                  ;Init the dest
3277 3E3A            LD      A,58
3279 EF              RST     4Ø
327A 28Ø7            JR      Z,LRLOK         ;If no error, cont.
327C FE2A            CP      42              ;Was it LRL error?
327E 28Ø3            JR      Z,LRLOK         ;Ignore if so
328Ø C39726          JP      EXIT3           ;  else real error, abort
3283 3A3F26   LRLOK  LD      A,(FCB2$+7)     ;P/u DEC of dest
3286 32FE32          LD      (DOFIL11+1),A
3289 ED4B2426        LD      BC,(FCB1$+12)   ;P/u ERN & ck for enuf
328D CDF333          CALL    WRERN           ; dest space on disk
329Ø C1              POP     BC              ;Recover DEC
3291 68              LD      L,B             ;Reset HL to dir
3292 262C            LD      H,BUF2$<-8
3294 C5              PUSH    BC              ;Save DEC
3295 28Ø6            JR      Z,DOFILØ2       ;Go if there was room
3297 CD1A27          CALL    PMTSRC          ;  else make source current, loop
329A C32A3Ø          JP      DONAM           ;  back because dest was swapped
329D 7D       DOFILØ2 LD     A,L             ;Check if date current
329E E6EØ            AND     ØEØH            ;Index to proper direc
32AØ 3C              INC     A
32A1 6F              LD      L,A
32A2 CB66            BIT     4,(HL)          ;Check if bit set
32A4 28Ø3            JR      Z,$+5
32A6 323A2F          LD      (SETBIT),A
               ;
32A9 21ØØØØ          LD      HL,Ø
```

```
32AC 224426    29Ø7Ø          LD      (FCB2$+12),HL    ;Set dest ERN to Ø
32AF           29Ø8Ø          @@REW                    ;Rewind the dest
32AF 3E44      ØØ234          LD      A,68
32B1 EF        ØØ235          RST     4Ø
32B2 2A1626    29Ø9Ø DOFILØ3  LD      HL,(BUFFER$)     ;Buffer addr
32B5 221B26    291ØØ DOFILØ4  LD      (FCB1$+3),HL     ;Set buffer addr in fcb
32B8 CD3D34    2911Ø          CALL    GETSRC           ;Prompt source & set fcb
32BB           2912Ø          @@READ                   ;Read a source file sector
32BB 3E43      ØØ236          LD      A,67
32BD EF        ØØ237          RST     4Ø
32BE 28ØB      2913Ø          JR      Z,DOFILØ5        ;Go if no error
32CØ FE1C      2914Ø          CP      1CH              ;Eof?
32C2 2824      2915Ø          JR      Z,DOFILØ9        ;Yes, finished loading
32C4 FE1D      2916Ø          CP      1DH              ;Nrn > ern?
32C6 282Ø      2917Ø          JR      Z,DOFILØ9        ;Also means load done
32C8 C39726    2918Ø          JP      EXIT3            ;Abort on any other error
32CB 24        2919Ø DOFILØ5  INC     H                ;Bump the buffer ptr
32CC 7C        292ØØ          LD      A,H
32CD FEØØ      2921Ø DOFILØ6  CP      $-$              ;Test out of memory
32CF 2ØE4      2922Ø          JR      NZ,DOFILØ4       ;Loop if more room
32D1 2A1626    2923Ø          LD      HL,(BUFFER$)     ;P/u buffer start
32D4 223B26    2924Ø DOFILØ7  LD      (FCB2$+3),HL     ;  & set into dest fcb
32D7 CD4634    2925Ø          CALL    GETDST           ;Prompt dest & set fcb
32DA           2926Ø          @@VER                    ;Write dest w/verify
32DA 3E49      ØØ238          LD      A,73
32DC EF        ØØ239          RST     4Ø
32DD C29726    2927Ø          JP      NZ,EXIT3         ;Quit on error
32EØ 24        2928Ø          INC     H                ;Bump buffer page
32E1 7C        2929Ø          LD      A,H
32E2 FEØØ      293ØØ DOFILØ8  CP      $-$              ;Out of memory?
32E4 2ØEE      2931Ø          JR      NZ,DOFILØ7       ;Write another if not
32E6 18CA      2932Ø          JR      DOFILØ3          ;  else back to loading
               2933Ø ;
               2934Ø ;        Reached the end of the source file
               2935Ø ;
32E8 CDD433    2936Ø DOFILØ9  CALL    LSTBUF           ;Write remaining buffer
32EB 2A2Ø26    2937Ø          LD      HL,(FCB1$+8)     ;P/u DEC & LRL
32EE 224Ø26    2938Ø          LD      (FCB2$+8),HL     ;  & stuff into dest
32F1 CD4634    2939Ø          CALL    GETDST           ;Set for dest fcb
32F4           294ØØ          @@CLOSE                  ;Close 'er up
32F4 3E3C      ØØ24Ø          LD      A,6Ø
32F6 EF        ØØ241          RST     4Ø
32F7 C29726    2941Ø          JP      NZ,EXIT3         ;Abort on close error
               2942Ø ;
               2943Ø ;        Now remove the mod flag from destination
               2944Ø ;         and do CLONE function
               2945Ø ;
32FA FD56Ø9    2946Ø          LD      D,(IY+9)         ;P/u dir cyl
32FD Ø6ØØ      2947Ø DOFIL11  LD      B,$-$            ;P/u DEC
32FF 78        2948Ø          LD      A,B              ;Pt to dir sector
33ØØ E61F      2949Ø          AND     1FH
33Ø2 C6Ø2      295ØØ          ADD     A,2              ;Bypass GAT and HIT
33Ø4 5F        2951Ø          LD      E,A
33Ø5 D5        2952Ø          PUSH    DE               ;Save cyl/sect
33Ø6 2A1626    2953Ø          LD      HL,(BUFFER$)     ;P/u buffer addr
33Ø9 CD7228    2954Ø          CALL    RDSEC            ;Read the dir sect
33ØC FEØ6      2955Ø          CP      6                ;Proper errcod?
33ØE 3E11      2956Ø          LD      A,17             ;Init "Dir read error
331Ø C29726    2957Ø          JP      NZ,EXIT3
```

Backup By Class

```
3313 78         2958Ø           LD      A,B             ;Pt to dir record
3314 E6EØ       2959Ø           AND     ØEØH
3316 5F         2960Ø           LD      E,A             ;Pt to DIR lo order
3317 3A1726     2961Ø           LD      A,(BUFFER$+1)   ;P/u hi order buffer pos
331A 57         2962Ø           LD      D,A
331B E1         2963Ø           POP     HL
331C C1         2964Ø           POP     BC              ;P/u DEC & buffer of src
331D C5         2965Ø           PUSH    BC
331E E5         2966Ø           PUSH    HL
331F 78         2967Ø           LD      A,B             ;Get source DEC
332Ø E6EØ       2968Ø           AND     ØEØH            ;  and pt to the direc
3322 6F         2969Ø           LD      L,A             ;  of the current file
3323 262C       2970Ø           LD      H,BUF2$<-8
3325 2C         2971Ø           INC     L               ;Pt to mod flag byte
3326 CBB6       2972Ø           RES     6,(HL)          ;Reset the MOD bit
3328 2D         2973Ø           DEC     L               ;Point to DIR+Ø
3329 Ø105ØØ     2974Ø           LD      BC,5            ;Transfer up thru
332C EDBØ       2975Ø           LDIR                    ;  DIR+4
332E 7B         2976Ø BYSPACE   LD      A,E             ;Point DE to the dest
332F C6ØB       2977Ø           ADD     A,11            ;  password fields
3331 5F         2978Ø           LD      E,A
3332 7D         2979Ø           LD      A,L             ;Point HL to the source
3333 C6ØB       2980Ø           ADD     A,11            ;  password fields
3335 6F         2981Ø           LD      L,A
3336 Ø104ØØ     2982Ø           LD      BC,4            ;Move both pswds
3339 EDBØ       2983Ø           LDIR
333B 2A1626     2984Ø           LD      HL,(BUFFER$)    ;P/u buffer addr
333E D1         2985Ø           POP     DE              ;Rcvr cyl/sect
333F CD6D28     2986Ø           CALL    WRSYS           ;Write back
3342 3E12       2987Ø           LD      A,18            ;Init "Dir write error
3344 C29726     2988Ø           JP      NZ,EXIT3        ;Quit on error
                2989Ø ;
                2990Ø ;         Attempt to clear mod flag of source
                2991Ø ;
3347 3EØØ       2992Ø DOFIL12   LD      A,Ø             ;Test for write prot src
3349 B7         2993Ø           OR      A               ;Which implies, can't
334A C22A2F     2994Ø           JP      NZ,SCNH1        ;  clear mod flags
334D C1         2995Ø           POP     BC              ;P/u DEC of source
334E C5         2996Ø           PUSH    BC
334F 78         2997Ø           LD      A,B             ;Clear mod flag on source
335Ø E6EØ       2998Ø           AND     ØEØH            ;Dir sector is resident
3352 3C         2999Ø           INC     A               ;In a buffer at BUF2
3353 6F         3ØØØØ           LD      L,A
3354 262C       3ØØ1Ø           LD      H,BUF2$<-8
3356 CBB6       3ØØ2Ø           RES     6,(HL)          ;Reset mod bit
3358 CD1A27     3ØØ3Ø           CALL    PMTSRC          ;Set for source i/o
335B FD56Ø9     3ØØ4Ø           LD      D,(IY+9)        ;P/u dir cyl
335E 78         3ØØ5Ø           LD      A,B             ;Pt to dir sect of source
335F E61F       3ØØ6Ø           AND     1FH
3361 C6Ø2       3ØØ7Ø           ADD     A,2             ;Adjust for GAT and HIT
3363 5F         3ØØ8Ø           LD      E,A
3364 21ØØ2C     3ØØ9Ø           LD      HL,BUF2$
3367 CD6D28     3Ø1ØØ           CALL    WRSYS           ;Write it back
336A CA2A2F     3Ø11Ø           JP      Z,SCNH1         ;Back on good write
336D FEØF       3Ø12Ø           CP      15              ;Accept only "write prot error
336F 3E12       3Ø13Ø           LD      A,18            ;Any other, "Dir write error
3371 C29726     3Ø14Ø           JP      NZ,EXIT3        ;  and quit
3374 3EFF       3Ø15Ø           LD      A,ØFFH          ;Turn off clear mod
3376 324833     3Ø16Ø           LD      (DOFIL12+1),A   ;  flag test
```

Backup By Class

```
3379                   @@LOGOT CCMOD$          ;"can't clear...
                       IFEQ    01H,1
3379 21C629            LD      HL,CCMOD$
                       ENDIF
337C 3E0C              LD      A,12
337E EF                RST     40
337F C32A2F            JP      SCNH1           ;Loop to next file
              ;
              ;        Routine to compare HL to DE, ret Z if equal
              ;
3382 7C       CPHLDE   LD      A,H             ;Test H=D
3383 92                SUB     D
3384 C0                RET     NZ              ;Back if not
3385 7D                LD      A,L             ;Test L=E
3386 93                SUB     E
3387 C9                RET                     ;Back with condition
              ;
              ;        Routine to construct filespec from name/ext
              ;
3388 3E3A     MAKSPC   LD      A,':'           ;Prepare for drivespec
338A 12                LD      (DE),A
338B 13                INC     DE
338C D5                PUSH    DE              ;Save pointer
338D 3A7B27            LD      A,(DSTDRV$+1)   ;P/u dest drive #
3390 E607              AND     7               ;Cvrt to ASCII
3392 C630              ADD     A,'0'
3394 12                LD      (DE),A          ; & stuff at filespec end
3395 13                INC     DE
3396 3E03              LD      A,3             ;Terminate with ETX
3398 12                LD      (DE),A
3399 211826            LD      HL,FCB1$        ;Copy source fcb to
339C 113826            LD      DE,FCB2$        ;  dest fcb
339F 012000            LD      BC,32
33A2 EDB0              LDIR
33A4 D1                POP     DE              ;Rcvr where source spec
33A5 3A0E27            LD      A,(SRCDRV$+1)   ;P/u source drive #
33A8 E607              AND     7               ;Cvrt to ASCII
33AA C630              ADD     A,'0'
33AC 12                LD      (DE),A          ;Stuff in dest fcb
33AD C9                RET
              ;
              ;        Routine to extract date from directory
              ;
33AE 7E       UNPACK   LD      A,(HL)          ;P/u DIR+1
33AF E60F              AND     0FH             ;Remove flags
33B1 57                LD      D,A             ;Save month
33B2 23                INC     HL              ;Pt to DIR+2
33B3 7E                LD      A,(HL)          ;P/u day and year
33B4 E6F8              AND     0F8H            ;Strip year
33B6 5F                LD      E,A             ;Save day in E
33B7 7E                LD      A,(HL)          ;Get the year back
33B8 AB                XOR     E               ;Strip the day
33B9 0F                RRCA                    ;Shift year to 5-7
33BA 0F                RRCA
33BB 0F                RRCA
33BC B2                OR      D               ;Merge with month
33BD 57                LD      D,A
33BE C9                RET
              ;
```

Backup By Class

```
                ;       Write the GAT back to disk
                ;
33BF 2EØØ       WRGAT   LD      L,Ø             ;HL to start of buffer
33C1 CD6D28             CALL    WRSYS           ;Write dir sector
33C4 3E15               LD      A,21            ;Init GAT write error
33C6 C29726             JP      NZ,EXIT3        ;  and quit on error
33C9 CD7728             CALL    VERSEC          ;Verify good write
33CC FEØ6               CP      6               ;Expect error 6
33CE 3E14               LD      A,2Ø            ;Init GAT read error
33DØ C29726             JP      NZ,EXIT3        ;Quit on any other error
33D3 C9                 RET
                ;
                ;       Write last buffer if needed
                ;
33D4 3A1726     LSTBUF  LD      A,(BUFFER$+1)   ;P/u hi order buffer start
33D7 BC                 CP      H               ;Are we there now?
33D8 C8                 RET     Z               ;Back if so, nothing loaded
33D9 3EØØ       LSTBUF1 LD      A,$-$           ;P/u last available page
33DB BC                 CP      H               ;There now?
33DC C8                 RET     Z               ;Already written if so
33DD 44                 LD      B,H             ;Need to write to this page
33DE 2A1626             LD      HL,(BUFFER$)    ;P/u buffer start
33E1 223B26     LSTBUF2 LD      (FCB2$+3),HL    ;  and put in dest fcb
33E4 CD4634             CALL    GETDST          ;Prompt dest
33E7                    @@VER                   ;Write with verify
33E7 3E49               LD      A,73
33E9 EF                 RST     4Ø
33EA C29726             JP      NZ,EXIT3        ;Quit on bad write
33ED 24                 INC     H               ;Bump buffer page
33EE 7C                 LD      A,H
33EF B8                 CP      B               ;At the end?
33FØ 2ØEF               JR      NZ,LSTBUF2      ;Loop if more
33F2 C9                 RET
                ;
                ;       Check if enough space on destination disk
                ;
33F3 78         WRERN   LD      A,B             ;If ERN = Ø, don't
33F4 B1                 OR      C               ; write a ERN
33F5 C8                 RET     Z
33F6 ØB                 DEC     BC              ;Adjust for Ø offset
33F7 CD4634             CALL    GETDST          ;Prompt dest
33FA D5                 PUSH    DE              ;Save fcb pointer
33FB                    @@POSN                  ;Position to end
33FB 3E42               LD      A,66
33FD EF                 RST     4Ø
33FE 2A1626             LD      HL,(BUFFER$)    ;P/u buffer addr
34Ø1 54                 LD      D,H             ;Construct a format
34Ø2 5D                 LD      E,L             ;  sector of all X'E5's
34Ø3 13                 INC     DE
34Ø4 Ø1FFØØ             LD      BC,255
34Ø7 36E5               LD      (HL),ØE5H
34Ø9 EDBØ               LDIR
34ØB D1                 POP     DE              ;Rcvr fcb ptr
34ØC                    @@VER                   ;Write with verify
34ØC 3E49               LD      A,73
34ØE EF                 RST     4Ø
34ØF C8                 RET     Z               ;Ret if no error
341Ø FE1B               CP      27              ;Disk Full?
3412 2Ø26               JR      NZ,NOTDF        ;No - quit on real error
```

Backup By Class

```
3414            3124Ø          @@REMOV                   ;Remove what can't fit
3414 3E39       ØØ253          LD      A,57
3416 EF         ØØ254          RST     4Ø
3417 FDCBØ35E 3125Ø            BIT     3,(IY+3)          ;Is this a rigid disk?
341B 28ØB       3126Ø          JR      Z,NOTHARD         ;Go if not
341D FDCBØ356 3127Ø            BIT     2,(IY+3)          ;Shown as Removable?
3421 28Ø5       3128Ø          JR      Z,NOTHARD         ;Prompt disk swap if so
3423 217D34     3129Ø          LD      HL,FULDRV$        ;Prepare disk full error
3426 183A       313ØØ          JR      DOING1
3428            3131Ø NOTHARD  @@FLAGS
3428 3E65       ØØ255          LD      A,1Ø1
342A EF         ØØ256          RST     4Ø
342B FDCB126E 3132Ø            BIT     5,(IY+'S'-'A')   ;Can't switch while DOing
342F 2Ø2E       3133Ø          JR      NZ,DOING
3431 218B34     3134Ø          LD      HL,NEWDISK        ;"disk full, enter new...
3434 CDD327     3135Ø          CALL    FLASH
3437 F6Ø1       3136Ø          OR      1                 ;Show switched dest
3439 C9         3137Ø          RET
343A            3138Ø NOTDF    EQU     $
343A C39726     3139Ø          JP      EXIT3             ;Error exit
                314ØØ ;
343D C5         3141Ø GETSRC   PUSH    BC
343E 111826     3142Ø          LD      DE,FCB1$          ;Pt to source FCB
3441 CD1A27     3143Ø          CALL    PMTSRC            ;Show source is current
3444 C1         3144Ø          POP     BC                ;  for disk I/O
3445 C9         3145Ø          RET
                3146Ø ;
3446 C5         3147Ø GETDST   PUSH    BC
3447 113826     3148Ø          LD      DE,FCB2$          ;Pt to dest FCB
344A CD8727     3149Ø          CALL    PMTDST            ;Show dest is current
344D C1         315ØØ          POP     BC                ;  for disk I/O
344E C9         3151Ø          RET
                3152Ø ;
344F FD56Ø9     3153Ø HITRD    LD      D,(IY+9)          ;P/u dir cyl of source
3452 1EØ1       3154Ø          LD      E,1               ;Read HIT
3454 21ØØ36     3155Ø          LD      HL,HITBUF         ;Into HIT buffer
3457 CD7228     3156Ø          CALL    RDSEC
345A FEØ6       3157Ø          CP      6                 ;Errcod correct?
345C 3E17       3158Ø          LD      A,17H             ;Init "HIT read error
345E C9         3159Ø          RET                       ;Return w/condition
                316ØØ ;
345F 21CA34     3161Ø DOING    LD      HL,DOMSG
3462 C3AF26     3162Ø DOING1   JP      EXIT4
                3163Ø ;
3465 1D         3164Ø CPYFIL$ DB       29,'Copying file: ',3
     43 6F 7Ø 79 69 6E 67 2Ø
     66 69 6C 65 3A 2Ø Ø3
3475 42         3165Ø QUERY    DB      'Backup ',3
     61 63 6B 75 7Ø 2Ø Ø3
347D 44         3166Ø FULDRV$ DB       'Disk is full ',CR
     69 73 6B 2Ø 69 73 2Ø 66
     75 6C 6C 2Ø ØD
348B 44         3167Ø NEWDISK DB       'Disk is full - Insert new formatted '
     69 73 6B 2Ø 69 73 2Ø 66
     75 6C 6C 2Ø 2D 2Ø 49 6E
     73 65 72 74 2Ø 6E 65 77
     2Ø 66 6F 72 6D 61 74 74
     65 64 2Ø
34AF 64         3168Ø          DB      'destination disk, <ENTER>',29,3
```

Backup By Class

```
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 64 69 73 6B 2C
     2Ø 3C 45 4E 54 45 52 3E
     1D Ø3
34CA 44          3169Ø DOMSG    DB       'Disk is full! - Can''t switch '
     69 73 6B 2Ø 69 73 2Ø 66
     75 6C 6C 21 2Ø 2D 2Ø 43
     61 6E 27 74 2Ø 73 77 69
     74 63 68 2Ø
34E7 77          317ØØ          DB       'while <DO> in effect',CR
     68 69 6C 65 2Ø 3C 44 4F
     3E 2Ø 69 6E 2Ø 65 66 66
     65 63 74 ØD
34FC 2Ø          3171Ø SIZBIG$ DB       '  File is larger than destination '
     2Ø 46 69 6C 65 2Ø 69 73
     2Ø 6C 61 72 67 65 72 2Ø
     74 68 61 6E 2Ø 64 65 73
     74 69 6E 61 74 69 6F 6E
     2Ø
351E 63          3172Ø          DB       'capacity - backup is bypassed',CR
     61 7Ø 61 63 69 74 79 2Ø
     2D 2Ø 62 61 63 6B 75 7Ø
     2Ø 69 73 2Ø 62 79 7Ø 61
     73 73 65 64 ØD
353C 43          3173Ø NOTSYS$ DB       'Can''t create SYSTEM disk - '
     61 6E 27 74 2Ø 63 72 65
     61 74 65 2Ø 53 59 53 54
     45 4D 2Ø 64 69 73 6B 2Ø
     2D 2Ø
3557 64          3174Ø          DB       'directory slots in use',CR
     69 72 65 63 74 6F 72 79
     2Ø 73 6C 6F 74 73 2Ø 69
     6E 2Ø 75 73 65 ØD
356E 7D          3175Ø QMARK$  DB       '} ? ',3
     2Ø 3F 2Ø Ø3
3573 4A          3176Ø MONTBL  DM       'JanFebMarAprMayJunJulAugSepOctNovDec'
     61 6E 46 65 62 4D 61 72
     41 7Ø 72 4D 61 79 4A 75
     6E 4A 75 6C 41 75 67 53
     65 7Ø 4F 63 74 4E 6F 76
     44 65 63
3597 A2          3177Ø SYSDEC  DB       ØA2H,ØC4H,2EH,2FH,2CH,2DH,2AH,2BH
     C4 2E 2F 2C 2D 2A 2B
359F 28          3178Ø          DB       28H,29H,26H,27H,27H,ØA7H,26H,ØA6H
     29 26 27 27 A7 26 A6
                 3179Ø ;
35A7 ØØ          318ØØ          DC       64,Ø             ;PATCH space
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
                 3181Ø ;
36ØØ             3182Ø          ORG      $<-8+1<+8
Ø1ØØ             3183Ø HITBUF   DS       256
                 3184Ø ;
```

```
37ØØ          3185Ø          SUBTTL  '<Backup Misc. routines>'
```

Backup Misc. routines

```
                ;
Ø9ØØ            CLSSIZ  EQU     $-BACKUP
                ;
                ;       Establish PC for rest of BACKUP initialization
                ;
41ØØ                    ORG     CORE$+MIRSIZ+CLSSIZ
41ØØ                    LORG    $               ;No offset here
                ;
                ;       Shift in Mirror or By-file module
                ;
41ØØ 3EØØ       CLSTST  LD      A,Ø             ;Non-zero if any option
41Ø2 B7                 OR      A
41Ø3 C21341             JP      NZ,MVBYCLS      ;Bypass if special
41Ø6 21ØØ32             LD      HL,MIRBU        ;Move in standard code
41Ø9 11ØØ2E             LD      DE,BACKUP
41ØC Ø1ØØØ6             LD      BC,MIRSIZ
41ØF EDBØ               LDIR
4111 1846               JR      SETBFR
                ;
4113 3ACF27     MVBYCLS LD      A,(SXORD+1)     ;Restrict by class
4116 B7                 OR      A               ; if a single drive
4117 2ØØ9               JR      NZ,MVBYC1
4119 211B44             LD      HL,CLS1DB$      ;Can't by class on 1 drv
411C            MOVNOT  @@DSPLY                 ;Display the error
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
411C 3EØA               LD      A,1Ø
411E EF                 RST     4Ø
411F C3AC26             JP      ABRTBU          ;  and abort the backup
                ;
4122 3ACA26     MVBYC1  LD      A,(XPARM$+1)    ;By class backup requires
4125 B7                 OR      A               ; either non (X) or residency
4126 2826               JR      Z,MVBYC2        ; of SYS 2, 3, 1Ø, and 12
4128 11ØØØØ     RESLOC  LD      DE,$-$          ;Store location (RES$)
412B 7B                 LD      A,E
412C B2                 OR      D               ;Check if there
412D 214744             LD      HL,RESREQ$      ;Init "Must be resident
413Ø 28EA               JR      Z,MOVNOT        ;Error if not in use
4132 D5                 PUSH    DE              ;OK, it's in use,
4133 DDE1               POP     IX              ;  are all modules
4135 DD7EØ9             LD      A,(IX+2*2+5)    ;  present and accounted
4138 B7                 OR      A               ;SYS2 resident?
4139 28E1               JR      Z,MOVNOT
413B DD7EØB             LD      A,(IX+3*2+5)    ;Is SYS3 resident?
413E B7                 OR      A
413F 28DB               JR      Z,MOVNOT
4141 DD7E19             LD      A,(IX+1Ø*2+5)   ;Is SYS1Ø resident?
4144 B7                 OR      A
4145 28D5               JR      Z,MOVNOT
4147 DD7E1D             LD      A,(IX+12*2+5)   ;Is SYS12 resident?
414A B7                 OR      A
414B CA1C41             JP      Z,MOVNOT
414E 21ØØ38     MVBYC2  LD      HL,CLSBU        ;Move in special code
4151 11ØØ2E             LD      DE,BACKUP
4154 Ø1ØØØ9             LD      BC,CLSSIZ
4157 EDBØ               LDIR
4159 1B         SETBFR  DEC     DE              ;Set the buffer
415A 14                 INC     D               ; one page above the code
415B 1EØØ               LD      E,Ø
```

Backup Misc. routines

```
415D ED531626 3242Ø          LD      (BUFFER$),DE     ;  and save starting posn
4161 C3ØØ2E   3243Ø          JP      BACKUP
              3244Ø ;
              3245Ø ;        Routine to get password
              3246Ø ;
4164 CD6D41   3247Ø GETMPW   CALL    GMPW1
4167 3EE4     3248Ø          LD      A,ØE4H           ;Get SYS2 for hash
4169 EF       3249Ø          RST     28H
              325ØØ ;
416A 3E84     3251Ø GETSYS2 LD       A,84H            ;Load SYS2, no function
416C EF       3252Ø          RST     28H
              3253Ø ;
416D 7A       3254Ø GMPW1    LD      A,D              ;Pswd entered as parm?
416E B3       3255Ø          OR      E
416F 281A     3256Ø          JR      Z,GMPW3          ;Prompt if not
4171 21ØØ2D   3257Ø          LD      HL,BUF3$
4174 E5       3258Ø          PUSH    HL
4175 Ø6Ø8     3259Ø          LD      B,8
4177 1A       326ØØ GMPW2    LD      A,(DE)           ;P/u pswd character
4178 FEØD     3261Ø          CP      CR               ;At end of line?
417A 282A     3262Ø          JR      Z,GMPW4          ;Space out if yes
417C FE2C     3263Ø          CP      ','              ;Comma separator?
417E 2826     3264Ø          JR      Z,GMPW4
418Ø FE22     3265Ø          CP      '"'              ;Closing quote?
4182 2822     3266Ø          JR      Z,GMPW4
4184 13       3267Ø          INC     DE
4185 77       3268Ø          LD      (HL),A           ;Xfer the character
4186 23       3269Ø          INC     HL
4187 1ØEE     327ØØ          DJNZ    GMPW2
4189 182Ø     3271Ø          JR      GMPW5
              3272Ø ;
              3273Ø ;        Not entered as parm, grab from keyboard
              3274Ø ;
418B          3275Ø GMPW3    @@DSPLY                  ;Display request
              ØØ262          IFEQ    ØØH,1
              ØØ263          LD      HL,
              ØØ264          ENDIF
418B 3EØA     ØØ265          LD      A,1Ø
418D EF       ØØ266          RST     4Ø
418E Ø1ØØØ8   3276Ø          LD      BC,8<8           ;Max 8 chars input
4191 21ØØ2D   3277Ø          LD      HL,BUF3$         ;Point to buffer
4194 E5       3278Ø          PUSH    HL
4195          3279Ø          @@KEYIN                  ;Grab password
4195 3EØ9     ØØ267          LD      A,9
4197 EF       ØØ268          RST     4Ø
4198 DAAC26   328ØØ          JP      C,ABRTBU         ;Abort on BREAK
419B EB       3281Ø          EX      DE,HL            ;Buf start to DE
419C 26ØØ     3282Ø          LD      H,Ø              ;Buf length to HL
419E 68       3283Ø          LD      L,B
419F 19       3284Ø          ADD     HL,DE            ;Pt to 1st unused pos
41AØ 3EØ8     3285Ø          LD      A,8              ;Calculate spaces needed
41A2 9Ø       3286Ø          SUB     B
41A3 28Ø6     3287Ø          JR      Z,GMPW5          ;Don't put any if 8 input
41A5 47       3288Ø          LD      B,A              ;Set space counter
41A6 362Ø     3289Ø GMPW4    LD      (HL),' '
41A8 23       329ØØ          INC     HL
41A9 1ØFB     3291Ø          DJNZ    GMPW4
41AB E1       3292Ø GMPW5    POP     HL               ;Rcvr pointer to buf
41AC E5       3293Ø          PUSH    HL
```

Backup Misc. routines

```
41AD Ø6Ø8      3294Ø         LD      B,8              ;Loop thru field
41AF 7E        3295Ø GMPW6   LD      A,(HL)
41BØ FE61      3296Ø         CP      'a'
41B2 38Ø6      3297Ø         JR      C,GMPW7
41B4 FE7B      3298Ø         CP      'z'+1
41B6 3ØØ2      3299Ø         JR      NC,GMPW7
41B8 CBAE      33ØØØ         RES     5,(HL)           ;Lc -> UC
41BA 23        33Ø1Ø GMPW7   INC     HL
41BB 1ØF2      33Ø2Ø         DJNZ    GMPW6
41BD D1        33Ø3Ø         POP     DE               ;Rcvr pointer to start
41BE C9        33Ø4Ø         RET
               33Ø5Ø ;
               33Ø6Ø ;       Check a drive for availability
               33Ø7Ø ;
               33Ø8Ø CKDRV
41BF 3AC827    33Ø9Ø         LD      A,(CURDSK+1)     ;P/u drive spec
41C2 4F        331ØØ         LD      C,A              ;Place in C
41C3 FD7EØØ    3311Ø         LD      A,(IY+Ø)         ;P/u drive vector
41C6 FEC3      3312Ø         CP      ØC3H             ;Ck for enabled
41C8 C24242    3313Ø         JP      NZ,CKDR5         ;Bypass if disabled
41CB E5        3314Ø         PUSH    HL
41CC D5        3315Ø         PUSH    DE
41CD FD7EØ6    3316Ø         LD      A,(IY+6)         ;Make sure the current
41DØ FDBEØ5    3317Ø         CP      (IY+5)           ;  cylinder count is in range
41D3 D2DC41    3318Ø         JP      NC,CKDRV1        ;Go if in range
41D6 CD5E28    3319Ø         CALL    RESTOR           ;Restore drive
41D9 C24F42    332ØØ         JP      NZ,CKDR7A        ;Go if error
               3321Ø ;
41DC FD56Ø5    3322Ø CKDRV1  LD      D,(IY+5)         ;P/u current track
41DF 1EØØ      3323Ø         LD      E,Ø              ;Set for sector Ø
41E1           3324Ø         @@SEEK                   ;Set track info to FDC
41E1 3E2E      ØØ269         LD      A,46
41E3 EF        ØØ27Ø         RST     4Ø
41E4 2Ø69      3325Ø         JR      NZ,CKDR7A        ;Go if error
41E6 CD6328    3326Ø         CALL    RSELCT           ;Wait until not busy
41E9 2Ø64      3327Ø         JR      NZ,CKDR7A        ;Not there - ret NZ
41EB FDCBØ35E  3328Ø         BIT     3,(IY+3)         ;If hard drive, bypass
41EF 2Ø47      3329Ø         JR      NZ,CKDR3A        ;  GAT data update
41F1 FDCBØ466  333ØØ         BIT     4,(IY+4)         ;If "ALIEN" by pass
41F5 2Ø1E      3331Ø         JR      NZ,CKDR2B        ;  test of index pulses
               3332Ø         IF      @MOD4
41F7 3EØ9      3333Ø         LD      A,Ø9             ;Set MSB of count down
41F9 F3        3334Ø         DI
               3335Ø         ENDIF
               3336Ø         IF      @MOD2
               3337Ø         LD      A,2Ø
               3338Ø         ENDIF
41FA 32Ø642    3339Ø INTRON  LD      (CDCNT+1),A      ;Store in 'LD H' instruction
41FD 212ØØØ    334ØØ         LD      HL,ØØ2ØH         ;Set up count (short)
               3341Ø ;
               3342Ø ;       Test for diskette in drive & rotating
               3343Ø ;
42ØØ CD4342    3344Ø CKDR1   CALL    INDEX            ;Test index pulse
42Ø3 2ØFB      3345Ø         JR      NZ,CKDR1         ;Jump on index
42Ø5 26ØØ      3346Ø CDCNT   LD      H,ØØH            ;CKDRV counter (long)
               3347Ø                                  ;Count set from above
42Ø7 CD4342    3348Ø CKDR2   CALL    INDEX            ;Test index pulse
42ØA 28FB      3349Ø         JR      Z,CKDR2          ;Jump on no index
               335ØØ         IF      @MOD4
```

Backup Misc. routines

```
420C FB         33510          EI                      ;OK for INTs now
                33520          ENDIF
420D 212000     33530          LD      HL,0020H        ;Index off wait (short)
4210 CD4342     33540 CKDR2A   CALL    INDEX
4213 20FB       33550          JR      NZ,CKDR2A       ;Jump on index
                33560 ;
                33570 ;        Diskette is rotating
                33580 ;
4215 F5         33590 CKDR2B   PUSH    AF              ;Save FDC status
4216 FD5609     33600          LD      D,(IY+9)
4219 210046     33610          LD      HL,CKDRBUF      ;Point to HIT buffer
421C 5D         33620          LD      E,L             ;Sector 0 for GAT
421D            33630          @@RDSSC                 ;Read the GAT
421D 3E55       00271          LD      A,85
421F EF         00272          RST     40
4220 202C       33640          JR      NZ,CKDR7        ;Jump on error
4222 2ACC46     33650          LD      HL,(CKDRBUF+0CCH)        ;P/u excess tracks
4225 3E22       33660          LD      A,22H           ;Add offset
4227 85         33670          ADD     A,L
4228 FD7706     33680          LD      (IY+6),A        ;Max track # to DCT
422B FDCB04AE   33690          RES     5,(IY+4)        ;Set to side 0
422F CB6C       33700          BIT     5,H             ;Test double sided
4231 2804       33710          JR      Z,CKDR3         ;Jump if only single
4233 FDCB04EE   33720          SET     5,(IY+4)        ;Set for side 2
4237 F1         33730 CKDR3    POP     AF              ;Recover FDC status
4238 07         33740 CKDR3A   RLCA                    ;Shift write prot to 7
4239 FDB603     33750          OR      (IY+3)          ;Merge Soft WP bit
423C E680       33760          AND     80H             ;Strip all but 7
423E 87         33770          ADD     A,A             ;Write prot to carry flg
                33780 ;
423F            33790 CKDR4    EQU     $
423F FB         33800          EI
4240 D1         33810          POP     DE
4241 E1         33820          POP     HL
4242 C9         33830 CKDR5    RET
4243 7C         33840 INDEX    LD      A,H             ;Count down tries
4244 B5         33850          OR      L
4245 2807       33860          JR      Z,CKDR7         ;Error if counted out
4247 2B         33870          DEC     HL              ;Dec the count
4248 CD6328     33880          CALL    RSELCT          ;Check for index pulse
424B CB4F       33890          BIT     1,A             ;Test index
424D C9         33900          RET                     ;Back with condition
424E F1         33910 CKDR7    POP     AF
424F 3E08       33920 CKDR7A   LD      A,8             ;Set Device not avail
4251 B7         33930          OR      A               ;Set NZ ret
4252 18EB       33940          JR      CKDR4           ;Leave
                33950 ;
                33960 ;        Data area
                33970 ;
                33980 PRMTBL$
0080            33990 VAL      EQU     80H
0040            34000 SW       EQU     40H
0020            34010 STR      EQU     20H
0010            34020 SGL      EQU     10H
4254 D3         34030          DB      'S'!80H
4255 63         34040          DB      SW!STR!3,'MPW',0
     4D 50 57   00
0005            34050 MPWRSP   EQU     $-PRMTBL$-1
425A DA30       34060          DW      MPWPRM
```

Backup Misc. routines

```
425C 73         34Ø7Ø           DB      SW!STR!SGL!3,'SYS',Ø
     53 59 53 ØØ
ØØØC            34Ø8Ø SYSRSP    EQU     $-PRMTBL$-1
4261 1Ø2F       34Ø9Ø           DW      SYSPRM+1
4263 53         341ØØ           DB      SW!SGL!3,'INV',Ø
     49 4E 56 ØØ
ØØ13            3411Ø INVRSP    EQU     $-PRMTBL$-1
4268 192F       3412Ø           DW      INVPRM+1
426A 53         3413Ø           DB      SW!SGL!3,'MOD',Ø
     4D 4F 44 ØØ
ØØ1A            3414Ø MODRSP    EQU     $-PRMTBL$-1
426F 1226       3415Ø           DW      MODPRM$
4271 55         3416Ø           DB      SW!SGL!5,'QUERY',Ø
     51 55 45 52 59 ØØ
ØØ23            3417Ø QRSP      EQU     $-PRMTBL$-1
4278 1426       3418Ø           DW      QPARM$
427A 41         3419Ø           DB      SW!1,'X',Ø
     58 ØØ
ØØ28            342ØØ XRSP      EQU     $-PRMTBL$-1
427D CA26       3421Ø           DW      XPARM$+1
427F 34         3422Ø           DB      STR!SGL!4,'DATE',Ø
     44 41 54 45 ØØ
ØØ3Ø            3423Ø DATRSP    EQU     $-PRMTBL$-1
4285 D82E       3424Ø           DW      DATPRM+1
4287 53         3425Ø           DB      SW!SGL!3,'NEW',Ø
     4E 45 57 ØØ
ØØ37            3426Ø NEWRSP    EQU     $-PRMTBL$-1
428C ØE26       3427Ø           DW      NEWPRM$
428E 53         3428Ø           DB      SW!SGL!3,'OLD',Ø
     4F 4C 44 ØØ
ØØ3E            3429Ø OLDRSP    EQU     $-PRMTBL$-1
4293 1Ø26       343ØØ           DW      OLDPRM$
4295 ØØ         3431Ø           NOP
                3432Ø ;
4296 53         3433Ø NOINDO$ DB        'Single drive backup invalid during'
     69 6E 67 6C 65 2Ø 64 72
     69 76 65 2Ø 62 61 63 6B
     75 7Ø 2Ø 69 6E 76 61 6C
     69 64 2Ø 64 75 72 69 6E
     67
42B8 2Ø         3434Ø           DB      ' <DO> processing',CR
     3C 44 4F 3E 2Ø 7Ø 72 6F
     63 65 73 73 69 6E 67 ØD
42C9 44         3435Ø NOFMT$  DB        'Destination disk not formatted'
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 64 69 73 6B 2Ø
     6E 6F 74 2Ø 66 6F 72 6D
     61 74 74 65 64
42E7 2Ø         3436Ø           DB      ' - Backup aborted',CR
     2D 2Ø 42 61 63 6B 75 7Ø
     2Ø 61 62 6F 72 74 65 64
     ØD
42F9 42         3437Ø HELLO$  DB        'BACKUP'
     41 43 4B 55 5Ø
42FF            3438Ø *GET    CLIENT:3
                3439Ø ;CLIENTS/ASM - File to establish sign-on headers
                344ØØ ;
42FF 2Ø         3441Ø         DB        ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
```

Backup Misc. routines

```
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
4329 2Ø          3442Ø          DB       ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                 3443Ø ;
433E 41          3444Ø          DB       'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
4366 2Ø          3445Ø          DB       ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
437E 43          3446Ø LDOS$    DB       'Command executes only from DOS Ready',CR
     6F 6D 6D 61 6E 64 2Ø 65
     78 65 63 75 74 65 73 2Ø
     6F 6E 6C 79 2Ø 66 72 6F
     6D 2Ø 44 4F 53 2Ø 52 65
     61 64 79 ØD
43A3 5Ø          3447Ø PRMERR$ DB        'Parameter error',CR
     61 72 61 6D 65 74 65 72
     2Ø 65 72 72 6F 72 ØD
43B3 53          3448Ø SRCNUM$ DB        'Source drive number ?        ',3
     6F 75 72 63 65 2Ø 64 72
     69 76 65 2Ø 6E 75 6D 62
     65 72 2Ø 3F 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø Ø3
43D1 44          3449Ø DSTNUM$ DB        'Destination drive number ?   ',3
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 64 72 69 76 65
     2Ø 6E 75 6D 62 65 72 2Ø
     3F 2Ø 2Ø 2Ø Ø3
43EF 4E          345ØØ NODAT$   DB       'No date established',CR
     6F 2Ø 64 61 74 65 2Ø 65
     73 74 61 62 6C 69 73 68
     65 64 ØD
44Ø3 42          3451Ø CLASS$   DB       'Backup by class invoked',CR
     61 63 6B 75 7Ø 2Ø 62 79
     2Ø 63 6C 61 73 73 2Ø 69
     6E 76 6F 6B 65 64 ØD
441B ØA          3452Ø CLS1DB$ DB        LF,'Single drive BACKUP invalid by files',CR
     53 69 6E 67 6C 65 2Ø 64
     72 69 76 65 2Ø 42 41 43
     4B 55 5Ø 2Ø 69 6E 76 61
     6C 69 64 2Ø 62 79 2Ø 66
     69 6C 65 73 ØD
                 3453Ø          IF       .NOT.SMALL
4441 53          3454Ø RES$     DB       'SYSRES'        ;Terminate with LF
     59 53 52 45 53
4447 ØA          3455Ø RESREQ$ DB        LF,'This backup requires residency '
     54 68 69 73 2Ø 62 61 63
     6B 75 7Ø 2Ø 72 65 71 75
     69 72 65 73 2Ø 72 65 73
```

Backup Misc. routines

```
     69 64 65 6E 63 79 2Ø
4467 6F         3456Ø          DB       'of SYS''s: 2, 3, 1Ø & 12.',CR
     66 2Ø 53 59 53 27 73 3A
     2Ø 32 2C 2Ø 33 2C 2Ø 31
     3Ø 2Ø 26 2Ø 31 32 2E ØD
                3457Ø          ENDIF
                3458Ø          IF       SMALL
                3459Ø RESREQ$ DB       'Backup by class requires the us'
                346ØØ          DB       'e of a SYSTEM diskette!       ',CR
                3461Ø          ENDIF
448Ø 42         3462Ø RECON$  DB       'Backup-reconstruct invoked',CR
     61 63 6B 75 7Ø 2D 72 65
     63 6F 6E 73 74 72 75 63
     74 2Ø 69 6E 76 6F 6B 65
     64 ØD
449B 43         3463Ø MIRROR$ DB       'Cylinder count differs - '
     79 6C 69 6E 64 65 72 2Ø
     63 6F 75 6E 74 2Ø 64 69
     66 66 65 72 73 2Ø 2D 2Ø
44B4 41         3464Ø          DB       'Attempt mirror-image backup ? ',3
     74 74 65 6D 7Ø 74 2Ø 6D
     69 72 72 6F 72 2D 69 6D
     61 67 65 2Ø 62 61 63 6B
     75 7Ø 2Ø 3F 2Ø Ø3
44D3 4D         3465Ø PMTMPW$ DB       'Master password ?      ',3
     61 73 74 65 72 2Ø 7Ø 61
     73 73 77 6F 72 64 2Ø 3F
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø Ø3
44EB 1F         3466Ø MAXDAYS DB       31,28,31,3Ø,31,3Ø,31,31,3Ø,31,3Ø,31
     1C 1F 1E 1F 1E 1F 1F 1E
     1F 1E 1F
44F7 42         3467Ø BADFMT$ DB       'Bad date format',CR
     61 64 2Ø 64 61 74 65 2Ø
     66 6F 72 6D 61 74 ØD
46ØØ            3468Ø CKDRBUF EQU      $<-8+1<8
Ø1ØØ            3469Ø          DS       256
46Ø7            347ØØ LAST    EQU      $
                ØØ14Ø ;
46Ø7            ØØ15Ø          SUBTTL <>
2EØØ            ØØ16Ø          END      BACKUP
```

```
$A1             2E9E $A2              2F1D $EX4             2EBB
@@1             ØØØØ @@2              ØØØØ @@3              ØØØØ
@@4             ØØØØ @MOD2            ØØØØ @MOD4            FFFF
ABRTBU          26AC ABRTBU$          2AØD ATTRIB           316C
AUTO            ØØEØ BACKUP           2EØØ BACKUPA          2EØ9
BADFMT          314E BADFMT$          44F7 BADMPW$          28E6
BCK1            2E2B BCK2             2E3F BCK3             2E5Ø
BCK4            2E6B BCK5             2E77 BCK6             2E9C
BOOTST$         26ØØ BREAK            26AC BSCLS            3218
BSMIR           3ØØ5 BUCAO$           29FC BUCORE$          2A69
BUF1$           2BØØ BUF2$            2CØØ BUF3$            2DØØ
BUFFER$         2616 BYCLAS           3ØAB BYPASS           3ØB5
BYSPACE         332E CANTBU           2FEC CANTBU$          2A1F
CCMOD$          29C6 CDCNT            42Ø5 CKBOOT           2FF7
CKCLA1          2F21 CKCLAS           2FØD CKDR1            42ØØ
CKDR2           42Ø7 CKDR2A           421Ø CKDR2B           4215
CKDR3           4237 CKDR3A           4238 CKDR4            423F
CKDR5           4242 CKDR7            424E CKDR7A           424F
CKDRBUF         46ØØ CKDRV            41BF CKDRV1           41DC
CKDST           3Ø5F CKGAT            2FF2 CKINV            2FB7
CKNAM           2FCØ CKNAMØ           2FCE CKNAM1           2FD8
CKNAM2          2FE5 CKOLD            3ØA2 CKSWDD           2889
CKTO            2EF8 CLASS$           44Ø3 CLS1DB$          441B
CLSBU           38ØØ CLSBUØ           2EA9 CLSBUØ1          2ECC
CLSBU1          2EDE CLSBU2           2EE4 CLSBU3           2EED
CLSBU4          2EF6 CLSBU5           2F1A CLSFLG$          2684
CLSSIZ          Ø9ØØ CLSTST           41ØØ CNTBAK1          312C
CORE$           32ØØ CPHLDE           3382 CPRID            2E36
CPRLOK          2EA6 CPYFIL$          3465 CPYMSG           3147
CR              ØØØD CURDSK           27C7 CVD1             31A8
CVD2            31AA CVD3             31BØ CVD7             31B9
CVTDEC          3196 CYL$             32Ø1 DAT              ØØD8
DATFLD$         2678 DATPRM           2ED7 DATRSP           ØØ3Ø
DIFDST$         298F DIFID$           3233 DIFSRC           276B
DIFSRC$         295D DIO1             287A DIRERR           2692
DOBU            3161 DOFILØ           323A DOFILØ2          329D
DOFILØ3         32B2 DOFILØ4          32B5 DOFILØ5          32CB
DOFILØ6         32CD DOFILØ7          32D4 DOFILØ8          32E2
DOFILØ9         32E8 DOFIL11          32FD DOFIL12          3347
DOING           345F DOING1           3462 DOMSG            34CA
DONAM           3Ø2A DONAM1           3Ø37 DONAM2           3Ø41
DONAM4          3Ø4F DONAM5           3Ø59 DOSYS1           31AB
DOSYS2          31FE DSTDFT           3Ø1A.DSTDIR           3ØØE
DSTDRV$         277A DSTNUM$          43D1 DSTWP$           28C1
DUCYL           2FA4 DUCYL$           31D7 DUCYL1           2FA6
DUCYL2          2FE1 DUCYL2A          3ØØB DUCYL2B          3ØØD
DUCYL3          3Ø1A DUCYL4           3Ø22 DUCYL5           3Ø25
DUCYL6          3Ø3Ø ERREXIT          26BA EX1              28B4
EX2             28BA EXIT             26BD EXIT1            2685
EXIT2           2695 EXIT3            2697 EXIT4            26AF
EXIT5           26C9 EXIT5A           26D8 EXIT5B           26DF
FCB1$           2618 FCB2$            2638 FCB3$            2658
FCNT1           1111 FCNT2            1555 FLASH            27D3
FLASHØ          27E3 FLS1             27F9 FLS2             2814
FLS4            2827 FLS5             2828 FLSH6            283D
FMPAKD$         268Ø FMT              ØØØØ FRCDAT           2FØ1
FRCPMT          27C1 FTFLG$           26Ø1 FULDRV$          347D
GETDAT          2F5D GETDAT1          2F7B GETDST           3446
```

```
GETGM           2F6D GETMPW          4164 GETSRC          343D
GETSYS2         416A GMPW1           416D GMPW2           4177
GMPW3           418B GMPW4           41A6 GMPW5           41AB
GMPW6           41AF GMPW7           41BA GODOIT          3ØA9
GOTDST          3Ø47 GOTSRC          2FAA HELLO$          42F9
HITBUF          36ØØ HITRD           344F IDMATCH         2E9A
INDEX           4243 INTRON          41FA INVPRM          2F18
INVRSP          ØØ13 LAST            46Ø7 LDCYL$          31C2
LDCYL2          2F7Ø LDCYL3          2F7A LDCYL4          2F7D
LDCYL5          2F81 LDCYL6          2F88 LDCYL7          2F8D
LDCYL8          2F9B LDOS$           437E LDTKS           2F3Ø
LDTKS1          2F37 LF              ØØØA LILBUF$         2658
LOCK            ØØ6Ø LRLOK           3283 LSTBUF          33D4
LSTBUF1         33D9 LSTBUF2         33E1 MAKSPC          3388
MATCHES         3Ø29 MAXDAYS         44EB MFLG$           26ØD
MIRBU           32ØØ MIRROR          3ØB7 MIRROR$         449B
MIRSIZ          Ø6ØØ MODPRM$         2612 MODRSP          ØØ1A
MONTBL          3573 MOVID           3Ø8F MOVNOT          411C
MPWPRM          3ØDA MPWRSP          ØØØ5 MVBYC1          4122
MVBYC2          414E MVBYCLS         4113 NDSYS$          26FB
NEWDISK         348B NEWPRM$         26ØE NEWRSP          ØØ37
NODAT$          43EF NODOIT          2FBD NOFMT$          42C9
NOINDO$         4296 NOPRMPT         3149 NOTDF           343A
NOTHARD         3428 NOTMIR          31BC NOTMIR$         32Ø5
NOTSYS          2EE7 NOTSYS$         353C OLDMPW          2E66
OLDMPW$         3272 OLDPRM$         261Ø OLDRSP          ØØ3E
OPENIT          2F28 PACKID$         3255 PACKNDO         2E97
PAKDAT          31ØB PARSDAT         3154 PASSWORD        42EØ
PMTDD           3Ø3D PMTDST          2787 PMTDST$         293A
PMTDST1         27BØ PMTMPW$         44D3 PMTSRC          271A
PMTSRC$         291C PMTSYS$         28FE PMTYN           2E7E
PMTYN$          32A4 PRMERR$         43A3 PRMTBL$         4254
PROT$           2A4A PROTSEC         2897 PRS4            318Ø
PRSD1           3159 PRSD2           3169 PRSD3           317E
PRSPEC          3ØFØ PS1             3ØFE PSRC1           275F
PSRC3           2774 PSWD            ØØCE QM1             3ØCØ
QMARK$          356E QPARM$          2614 QRSP            ØØ23
QUERY           3475 RDBOOT          31EB RDSEC           2872
RECON           3ØAE RECON$          448Ø RES$            4441
RESKFLG         284C RESLOC          4128 RESMF           3ØDØ
RESMF1          3ØDC RESMF2          3ØFØ RESMF2A         314E
RESMF2B         3157 RESMF3          3162 RESMF4          316E
RESMF5          318D RESMF6          318F RESREQ$         4447
RESTOR          285E RETCOD          26C1 RSELCT          2863
SCDAT1          3ØD2 SCDAT2          3ØE7 SCDAT4          311D
SCNH1           2F2A SCNH2           2F2E SCNH3           2F5C
SCNH4           2F9C SCNH4A          2FA9 SCNH5           2FF9
SCNH6           2FFA SCNH7           3Ø15 SCNHIT          2F29
SELECT          2859 SETØ            3ØØ8 SETBFR          4159
SETBIT          2F3A SETSYS          2EA2 SGL             ØØ1Ø
SHOPROT         3Ø8F SIZBIG$         34FC SIZOK           3269
SIZSAV          3258 SMALL           ØØØØ SPCFLD$         26Ø2
SPSAV           26BD SRCDFT          2F95 SRCDRV$         27ØD
SRCNUM$         43B3 STR             ØØ2Ø STRDIR$         31Ø3
SVCTR           28A1 SW              ØØ4Ø SXORD           27CE
SYSDEC          3597 SYSDRV$         27ØØ SYSPRM          2FØF
SYSRSP          ØØØC TKCAP           ØØCC TOEXIT1         2F59
TOPAKD$         2682 TST5_8          3Ø4C TSTCAP          3Ø97
TSTDRV          2855 TSTMFLG         2FF2 TSTMPW          3ØCF
```

```
UNPACK          33AE VAL             ØØ8Ø VECYL$          31EC
VECYL1          3Ø3C VECYL2          3Ø6E VECYL3          3Ø78
VECYL4          3Ø7A VECYL5          3Ø7E VECYL6          3Ø84
VERSEC          2877 VRBOOT          3232 WRBOOT          32Ø9
WRBOOT1         321B WRBOOT2         3223 WRERN           33F3
WRGAT           33BF WRSEC           2868 WRSYS           286D
XPARM$          26C9 XRSP            ØØ28 @@ABORT         6C65
@@ADTSK         6CF8 @@BANK          721Ø @@BKSP          6EFØ
@@BREAK         7226 @@CHNIO         6C5Ø @@CKBRKC        7274
@@CKDRV         6D4C @@CKEOF         6FØ5 @@CKTSK         6CE3
@@CLOSE         6EDB @@CLS           725E @@CMNDI         6C8F
@@CMNDR         6CA4 @@CTL           6AB4 @@DATE          6C26
@@DCSTAT        6D8B @@DEBUG         6CCE @@DECHEX        719Ø
@@DIRRD         7ØFD @@DIRWR         7112 @@DIV16         717B
@@DIV8          7166 @@DODIR         6D61 @@DSP           6A78
@@DSPLY         6B18 @@ERROR         6CB9 @@EXIT          6C7A
@@FEXT          7Ø6A @@FLAGS         71FA @@FNAME         7Ø7F
@@FSPEC         7Ø55 @@GATRD         7ØE8 @@GATWR         7127
@@GET           6A8C @@GTDCB         7ØA9 @@GTDCT         7Ø94
@@GTMOD         7ØBE @@HDFMT         6E33 @@HEX16         71CF
@@HEX8          71BA @@HEXDEC        71A5 @@HIGH$         71E4
@@INIT          6EB1 @@KBD           6AFØ @@KEY           6A64
@@KEYIN         6BØ4 @@KLTSK         6D37 @@LOAD          7Ø2B
@@LOC           6F1A @@LOF           6F2F @@LOGER         6B4F
@@LOGOT         6B64 @@MSG           6B9B @@MUL16         7151
@@MUL8          713C @@OPEN          6EC6 @@PARAM         6C11
@@PAUSE         6BFC @@PEOF          6F44 @@POSN          6F59
@@PRINT         6BBØ @@PRT           6AC8 @@PUT           6AAØ
@@RAMDIR        6D76 @@RDSEC         6EØ9 @@RDSSC         7ØD3
@@READ          6F6E @@REMOV         6E9C @@RENAM         6E87
@@REW           6F83 @@RMTSK         6DØD @@RPTSK         6D22
@@RREAD         6F98 @@RSLCT         6DF4 @@RSTOR         6DB5
@@RUN           7Ø4Ø @@RWRIT         6FAD @@SEEK          6DDF
@@SEEKSC        6FC2 @@SKIP          6FD7 @@SLCT          6DAØ
@@STEPI         6DCA @@TIME          6C3B @@VDCTL         6BE7
@@VER           6FEC @@VRSEC         6E1E @@WEOF          7ØØ1
@@WHERE         6ADC @@WRITE         7Ø16 @@WRSEC         6E48
@@WRSSC         6E5D @@WRTRK         6E72
2EØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

CLICK/FLT - Sound click device filter

The Click filter can be used to generate a short clicking sound on the occurence of all characters sent to a device, or on a specific character only. Click will always install itself in high memory, and will not attempt to load in the low driver zone. It is installed with the SET and FILTER Library commands.

```
               ;CLICK/ASM - Device Click Filter
0000                    TITLE   <CLICK/FLT - LS-DOS 6.2>
               ;
               ;
                        IF      @MOD4
0048           TONE     EQU     48H
0018           LEN      EQU     18H
0090           SNDPORT  EQU     90H
                        ENDIF
                        IF      @MOD2
               LEN      EQU     180H                ;Length
               SNDPORT  EQU     0A0H
                        ENDIF
               ;
0000           *GET     SVCMAC:3                    ;SVC Macro equivalents
               ;SVCMAC/ASM - LS-DOS Version VI
               *LIST    OFF
               *LIST    ON
0000           *GET     VALUES:3                    ;Misc. equates
               ;VALUES/ASM - Version 6
               *LIST OFF
               *LIST ON
0000           *GET     COPYCOM:3                   ;Copyright messages
               ; COPYCOM - File for Copyright COMment block
               ;
0000                    COM     '<*(C) 1982,83,84 by LSI*>'
               ;
2400                    ORG     2400H
               ;
               START
2400                    @@CKBRKC
2400 3E6A               LD      A,106
2402 EF                 RST     40
2403 2804               JR      Z,STARTA            ;Continue if no BREAK
2405 21FFFF             LD      HL,-1               ; set up abort RET
2408 C9                 RET
               ;
2409 ED730B25  STARTA   LD      (EXIT+1),SP         ;Save stack for error exit
240D CD1724             CALL    DOINIT              ;Do initialization
2410 CD9624             CALL    INSTFLT             ;Relocate/install filter
2413 210000    NORMEX   LD      HL,0                ;Good exit
2416 C9                 RET
               ;
               ;        Xfer DCB ptr to IX & stuff addrs' in driver
               ;
2417 D5        DOINIT   PUSH    DE                  ;DE => DCB+0
2418 DDE1               POP     IX                  ;Xfer to IX
241A ED535F24           LD      (DCB),DE            ;Xfer into header
               ;
               ;        Sign-on
               ;
241E E5                 PUSH    HL
241F 215125             LD      HL,HELLO$           ;Sign on message
2422 CDEC24             CALL    DSPLY
               ;
               ;        Check PARMS and if entry from SET command
               ;
2425 111125             LD      DE,PRMTBL           ;Point to parms
2428 E1                 POP     HL                  ;Recover cmdline posn
2429                    @@PARAM                     ;Parse the parms
2429 3E11               LD      A,17
```

```
242B EF         ØØØØ4            RST     4Ø
242C C2F224     ØØ59Ø            JP      NZ,IOERR          ;Exit on parm error
                ØØ6ØØ ;
242F            ØØ61Ø            @@FLAGS                   ;IY => System Flags Base
242F 3E65       ØØØØ5            LD      A,1Ø1
2431 EF         ØØØØ6            RST     4Ø
2432 FDCBØ25E   ØØ62Ø            BIT     3,(IY+'C'-'A')    ;System request?
2436 CAFD24     ØØ63Ø            JP      Z,VIASET          ;"Install with SET
                ØØ64Ø ;
                ØØ65Ø ;          Before anything - Make sure hi-mem is avail
                ØØ66Ø ;
2439 FDCBØ246   ØØ67Ø            BIT     Ø,(IY+CFLAG$)     ;High memory available ?
243D C2Ø125     ØØ68Ø            JP      NZ,CANT           ;No - display error
                ØØ69Ø ;
                ØØ7ØØ ;          Set up filter for CHAR if entered
                ØØ71Ø ;
244Ø 11ØØØØ     ØØ72Ø CHARPRM    LD      DE,ØØ             ;Char parm lands here
2443 7A         ØØ73Ø            LD      A,D               ;Check if entered and
2444 BB         ØØ74Ø            CP      E                 ;  is normal character
2445 C8         ØØ75Ø            RET     Z                 ;Done if not entered
2446 FEØØ       ØØ76Ø            CP      Ø                 ;Check is MSB is altered
2448 3E2C       ØØ77Ø            LD      A,44              ;Init "Parameter error
244A C2F224     ØØ78Ø            JP      NZ,IOERR          ;Bad if so
                ØØ79Ø ;
244D 53         ØØ8ØØ            LD      D,E               ;Set up CP nn
244E 1EFE       ØØ81Ø            LD      E,ØFEH            ;Reverse it and
245Ø ED537424   ØØ82Ø            LD      (CKCHAR),DE       ; put it in the filter
2454 C9         ØØ83Ø            RET
                ØØ84Ø ;*=*=*
                ØØ85Ø ;          Actual CLICK filter Code
                ØØ86Ø ;*=*=*
2455 18ØC       ØØ87Ø HEADER     JR      FILTER
2457 ØØØØ       ØØ88Ø OLDHI      DW      Ø                 ;HIGH$ before CLICK
2459 Ø5         ØØ89Ø            DB      5,'CLICK'
     43 4C 49 43 4B
245F ØØØØ       ØØ9ØØ DCB        DW      $-$               ;DCB pointing to CLICK
2461 ØØØØ       ØØ91Ø SPARE      DW      Ø                 ;System wants it
                ØØ92Ø ;
                ØØ93Ø ;          Is there a character here?
                ØØ94Ø ;
2463 DD2A5F24   ØØ95Ø FILTER     LD      IX,(DCB)          ;P/u DCB address
2467 38Ø6       ØØ96Ø            JR      C,NOTCTL          ;Go if Get
2469 28Ø4       ØØ97Ø            JR      Z,NOTCTL          ;  or Put
246B            ØØ98Ø IS_CTL     @@CHNIO                   ;Pass the CTL call
246B 3E14       ØØØØ7            LD      A,2Ø
246D EF         ØØØØ8            RST     4Ø
246E C9         ØØ99Ø            RET
246F            Ø1ØØØ NOTCTL     @@CHNIO                   ;Go to next in line
246F 3E14       ØØØØ9            LD      A,2Ø
2471 EF         ØØØ1Ø            RST     4Ø
2472 CØ         Ø1Ø1Ø            RET     NZ                ;None - RETurn NZ
                Ø1Ø2Ø ;
                Ø1Ø3Ø ;          Generate short Click
                Ø1Ø4Ø ;
2473 F5         Ø1Ø5Ø SOUND      PUSH    AF                ;Save registers
2474 ØØØØ       Ø1Ø6Ø CKCHAR     DW      ØØ                ;Space for a CP instruct
2476 2Ø1C       Ø1Ø7Ø            JR      NZ,POPAF          ; exit if CP above fails
2478 C5         Ø1Ø8Ø SNDNOW     PUSH    BC
2479 D5         Ø1Ø9Ø            PUSH    DE
                Ø11ØØ            IF      @MOD2
                Ø111Ø            LD      BC,LEN            ;D-ration
```

```
               Ø112Ø         LD      A,-1            ;ON value
               Ø113Ø         OUT     (SNDPORT),A     ;Turn on sound
               Ø114Ø         LD      A,16            ;Svc @PAUSE
               Ø115Ø         RST     28H             ;Delay
               Ø116Ø         XOR     A               ;OFF value
               Ø117Ø         OUT     (SNDPORT),A     ;Turn off sound
               Ø118Ø         ENDIF
               Ø119Ø ;
               Ø12ØØ         IF      @MOD4
               Ø121Ø ;
247A 111848    Ø122Ø STFVALS LD      DE,TONE<8!LEN   ;D = Tone, E = Length
247D 3EØØ      Ø123Ø         LD      A,Ø             ;Init on/off toggle
247F ØE9Ø      Ø124Ø         LD      C,SNDPORT       ;Point to port
               Ø125Ø ;
               Ø126Ø ;       ON portion
               Ø127Ø ;
2481 3C        Ø128Ø DURLP   INC     A               ;Hold output high
2482 ED79      Ø129Ø         OUT     (C),A           ;  for count of (B)
2484 42        Ø13ØØ         LD      B,D             ;Play tone
2485 1ØFE      Ø131Ø         DJNZ    $
               Ø132Ø ;
               Ø133Ø ;OFF portion
               Ø134Ø ;
2487 3D        Ø135Ø         DEC     A               ;  for count of (B)
2488 ED79      Ø136Ø         OUT     (C),A
248A 42        Ø137Ø         LD      B,D             ;Hold output low for
248B 1ØFE      Ø138Ø         DJNZ    $
               Ø139Ø ;
248D 1D        Ø14ØØ         DEC     E               ;Dec the duration
248E 2ØF1      Ø141Ø         JR      NZ,DURLP
249Ø 1ØFE      Ø142Ø         DJNZ    $               ;Hold for 256 count
               Ø143Ø         ENDIF
               Ø144Ø ;
2492 D1        Ø145Ø         POP     DE              ;Restore regs
2493 C1        Ø146Ø         POP     BC
2494 F1        Ø147Ø POPAF   POP     AF
2495 C9        Ø148Ø         RET                     ;And RETurn
               Ø149Ø ;
ØØ41           Ø15ØØ LENGTH  EQU     $-HEADER        ;Length of Filter
               Ø151Ø ;
               Ø152Ø ;       INSTFLT - Relocate & Install Filter
               Ø153Ø ;
2496 DD36ØØ47  Ø154Ø INSTFLT LD      (IX+Ø),47H      ;Set Filter,Ctl,Get,Put
               Ø155Ø ;
               Ø156Ø ;       Pick up Old HIGH$ and save in driver
               Ø157Ø ;
249A 21ØØØØ    Ø158Ø         LD      HL,Ø            ;Get HIGH$
249D 45        Ø159Ø         LD      B,L
249E           Ø16ØØ         @@HIGH$
249E 3E64      ØØØ11         LD      A,1ØØ
24AØ EF        ØØØ12         RST     4Ø
24A1 225724    Ø161Ø         LD      (OLDHI),HL      ;Stuff into header
               Ø162Ø ;
               Ø163Ø ;       Calculate New HIGH$ & stuff into DCB
               Ø164Ø ;
24A4 Ø141ØØ    Ø165Ø         LD      BC,LENGTH       ;Length of driver
24A7 C5        Ø166Ø         PUSH    BC              ;Save length
24A8 B7        Ø167Ø         OR      A
24A9 ED42      Ø168Ø         SBC     HL,BC           ;HL => New HIGH$
24AB           Ø169Ø         @@HIGH$                 ;(B=Ø) set new HIGH$
24AB 3E64      ØØØ13         LD      A,1ØØ
```

```
24AD EF                 RST     4Ø
24AE 23                 INC     HL              ;Pt to driver
24AF DD75Ø1             LD      (IX+1),L        ;Stuff driver address
24B2 DD74Ø2             LD      (IX+2),H        ;  into DCB
                ;
                ;       Calc offset between source & dest for relo
                ;
24B5 115524             LD      DE,HEADER       ;Start of driver
24B8 E5                 PUSH    HL              ;Save Source & Dest ptrs
24B9 D5                 PUSH    DE
24BA B7                 OR      A               ;Clear carry
24BB ED52               SBC     HL,DE           ;Get offset
                ;
                ;       Relocate internal references in driver
                ;
24BD DD21DC24           LD      IX,RELTBL       ;Point to relocation tbl
24C1 44                 LD      B,H             ;Move to BC
24C2 4D                 LD      C,L
24C3 DD6EØØ     RLOOP   LD      L,(IX)          ;Get address to change
24C6 DD66Ø1             LD      H,(IX+1)
24C9 7C                 LD      A,H
24CA B5                 OR      L
24CB 2819               JR      Z,RELDUN
24CD 5E                 LD      E,(HL)          ;P/U address
24CE 23                 INC     HL
24CF 56                 LD      D,(HL)
24DØ EB                 EX      DE,HL           ;Offset it
24D1 Ø9                 ADD     HL,BC
24D2 EB                 EX      DE,HL
24D3 72                 LD      (HL),D          ;Put it back
24D4 2B                 DEC     HL
24D5 73                 LD      (HL),E
24D6 DD23               INC     IX
24D8 DD23               INC     IX
24DA 18E7               JR      RLOOP           ;Loop till done
                ;
                ;       Relocation Table for Driver
                ;
24DC 6524       RELTBL  DW      FILTER+2,Ø,Ø,Ø,Ø
     ØØØØ ØØØØ ØØØØ ØØØØ
                ;
                ;       Transfer Filter code to high memory
                ;
24E6 E1         RELDUN  POP     HL              ;HL => Source DE => Dest
24E7 D1                 POP     DE
24E8 C1                 POP     BC              ;BC = length of filter
24E9 EDBØ               LDIR                    ;Block move
24EB C9                 RET                     ;RETurn
                ;
                ;       DSPLY - Display a string
                ;
24EC D5         DSPLY   PUSH    DE              ;Save DE
24ED                    @@DSPLY                 ;Display it
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
24ED 3EØA               LD      A,1Ø
24EF EF                 RST     4Ø
24FØ D1                 POP     DE              ;
24F1 C8                 RET     Z               ;Return if good
                ;
                02240 ;         IOERR - Any fatal Errors come here
                02250 ;
24F2 6F         02260 IOERR    LD      L,A             ;Xfer error # to HL
24F3 2600       02270          LD      H,0             ;
24F5 F6C0       02280          OR      0C0H            ;Short msg & RETurn
24F7 4F         02290          LD      C,A
24F8            02300          @@ERROR                 ;Display error
24F8 3E1A       00020          LD      A,26
24FA EF         00021          RST     40
24FB 180D       02310          JR      EXIT            ;Go to exit routine
                02320 ;
                02330 ;        Error Handler
                02340 ;
24FD 213C25     02350 VIASET   LD      HL,VIASET$      ;"Install with Set
2500 DD         02360          DB      0DDH
2501 212225     02370 CANT     LD      HL,CANT$        ;"No memory space
                02380 ;
2504            02390          @@LOGOT                 ;Log error
                00022          IFEQ    00H,1
                00023          LD      HL,
                00024          ENDIF
2504 3E0C       00025          LD      A,12
2506 EF         00026          RST     40
2507 21FFFF     02400          LD      HL,-1           ;Set abort code
                02410 ;
250A 310000     02420 EXIT     LD      SP,$-$          ;P/u original SP
250D            02430          @@CKBRKC                ;Clear out break
250D 3E6A       00027          LD      A,106
250F EF         00028          RST     40
2510 C9         02440          RET                     ;  and RETurn
                02450 ;
2511 43         02460 PRMTBL   DB      'CHAR  '
     48 41 52 20 20
2517 4124       02470          DW      CHARPRM+1
2519 43         02480          DB      'C     '
     20 20 20 20 20
251F 4124       02490          DW      CHARPRM+1
2521 00         02500          NOP                     ;End of table
                02510 ;
                02520 ;
2522 4E         02530 CANT$    DB      'No memory space available',CR
     6F 20 6D 65 6D 6F 72 79
     20 73 70 61 63 65 20 61
     76 61 69 6C 61 62 6C 65
     0D
253C 4D         02540 VIASET$ DB       'Must install via SET',CR
     75 73 74 20 69 6E 73 74
     61 6C 6C 20 76 69 61 20
     53 45 54 0D
                02550 ;
2551 43         02560 HELLO$  DB       'CLICK'
     4C 49 43 4B
2556            02570 *GET    CLIENT:3
                04240 ;CLIENTS/ASM - File to establish sign-on headers
                04250 ;
2556 20         04260         DB       ' - 6.2.0 - Copyright 1982/83/84 by Logical'
     2D 20 36 2E 32 2E 30 20
     2D 20 43 6F 70 79 72 69
     67 68 74 20 31 39 38 32
     2F 38 33 2F 38 34 20 62
     79 20 4C 6F 67 69 63 61
     6C
258Ø 2Ø         Ø427Ø          DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                Ø428Ø ;
2595 41         Ø429Ø          DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
25BD 2Ø         Ø43ØØ          DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
                Ø258Ø ;
24ØØ            Ø259Ø          END     START
```

```
@@1             ØØØØ @@2              ØØØØ @@3              ØØØØ
@@4             ØØØØ @MOD2            ØØØØ @MOD4            FFFF
ABB             ØØ1Ø AP               ØØ27 BREAK            ØØ8Ø
BS              ØØØ8 CANT             25Ø1 CANT$            2522
CFLAG$          ØØØ2 CHARPRM          244Ø CKCHAR           2474
CR              ØØØD DCB              245F DFLAG$           ØØØ3
DOINIT          2417 DSPLY            24EC DURLP            2481
ETX             ØØØ3 EXIT             25ØA FILTER           2463
FLAG            ØØ4Ø HEADER           2455 HELLO$           2551
INSTFLT         2496 IOERR            24F2 IS_CTL           246B
KFLAG$          ØØØA LEN              ØØ18 LENGTH           ØØ41
LF              ØØØA NORMEX           2413 NOTCTL           246F
NUM             ØØ8Ø OLDHI            2457 PAR_ERR          ØØ2C
POPAF           2494 PRMTBL           2511 RELDUN           24E6
RELTBL          24DC RLOOP            24C3 SFLAG$           ØØ12
SNDNOW          2478 SNDPORT          ØØ9Ø SOUND            2473
SPARE           2461 START            24ØØ STARTA           24Ø9
STFVALS         247A STR              ØØ2Ø TAB              ØØØ9
TONE            ØØ48 VFLAG$           ØØ15 VIASET           24FD
VIASET$         253C @@ABORT          8Ø2Ø @@ADTSK          8ØB3
@@BANK          85CB @@BKSP           82AB @@BREAK          85E1
@@CHNIO         8ØØB @@CKBRKC         862F @@CKDRV          81Ø7
@@CKEOF         82CØ @@CKTSK          8Ø9E @@CLOSE          8296
@@CLS           8619 @@CMNDI          8Ø4A @@CMNDR          8Ø5F
@@CTL           7E6F @@DATE           7FE1 @@DCSTAT         8146
@@DEBUG         8Ø89 @@DECHEX         854B @@DIRRD          84B8
@@DIRWR         84CD @@DIV16          8536 @@DIV8           8521
@@DODIR         811C @@DSP            7E33 @@DSPLY          7ED3
@@ERROR         8Ø74 @@EXIT           8Ø35 @@FEXT           8425
@@FLAGS         85B5 @@FNAME          843A @@FSPEC          841Ø
@@GATRD         84A3 @@GATWR          84E2 @@GET            7E47
@@GTDCB         8464 @@GTDCT          844F @@GTMOD          8479
@@HDFMT         81EE @@HEX16          858A @@HEX8           8575
@@HEXDEC        856Ø @@HIGH$          859F @@INIT           826C
@@KBD           7EAB @@KEY            7E1F @@KEYIN          7EBF
@@KLTSK         8ØF2 @@LOAD           83E6 @@LOC            82D5
@@LOF           82EA @@LOGER          7FØA @@LOGOT          7F1F
@@MSG           7F56 @@MUL16          85ØC @@MUL8           84F7
@@OPEN          8281 @@PARAM          7FCC @@PAUSE          7FB7
@@PEOF          82FF @@POSN           8314 @@PRINT          7F6B
@@PRT           7E83 @@PUT            7E5B @@RAMDIR         8131
@@RDSEC         81C4 @@RDSSC          848E @@READ           8329
@@REMOV         8257 @@RENAM          8242 @@REW            833E
@@RMTSK         8ØC8 @@RPTSK          8ØDD @@RREAD          8353
@@RSLCT         81AF @@RSTOR          817Ø @@RUN            83FB
@@RWRIT         8368 @@SEEK           819A @@SEEKSC         837D
@@SKIP          8392 @@SLCT           815B @@STEPI          8185
@@TIME          7FF6 @@VDCTL          7FA2 @@VER            83A7
@@VRSEC         81D9 @@WEOF           83BC @@WHERE          7E97
@@WRITE         83D1 @@WRSEC          82Ø3 @@WRSSC          8218
@@WRTRK         822D
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

COM/DVR - RS232 hardware driver

The Com driver program will initialize the UART and allow characters to be sent and received via the RS232 hardware. The driver will attempt to install itself in the low driver zone, but will relocate to high memory if necessary. It must be installed with the SET Library command.

```
                 ;COM/ASM - RS232 Driver Program
ØØØØ                     TITLE   '<COM/DVR - LS-DOS 6.2>'
                 ;
ØØØA             LF      EQU     1Ø
ØØØD             CR      EQU     13
                 ;
ØØØØ             *GET    COPYCOM:3                ;Copyright message
                 ; COPYCOM - File for Copyright COMment block
                 ;
ØØØØ                     COM     '<*(C) 1982,83,84 by LSI*>'
ØØØØ             *GET    SVCMAC:3                 ;SVC Macro equivalents
                 ;SVCMAC/ASM - LS-DOS Version VI
                 *LIST   OFF
                 *LIST   ON
                 ;
24ØØ                     ORG     24ØØH
                 ;
                 BEGIN
24ØØ                     @@CKBRKC
24ØØ 3E6A                LD      A,1Ø6
24Ø2 EF                  RST     4Ø
24Ø3 28Ø4                JR      Z,BEGINA         ;Continue if no BREAK
24Ø5 21FFFF              LD      HL,-1
24Ø8 C9                  RET                      ;Return with abort code
                 ;
24Ø9 D5          BEGINA  PUSH    DE               ;Save DCB address
24ØA DDE1                POP     IX               ;  in index reg
24ØC ED537F26            LD      (CLDCB),DE       ;  and in driver header
241Ø                     @@DSPLY HELLO$           ;Welcome the user
                         IFEQ    Ø1H,1
241Ø 215B25              LD      HL,HELLO$
                         ENDIF
2413 3EØA                LD      A,1Ø
2415 EF                  RST     4Ø
                 ;
                 ;       Check if entry from SET command
                 ;
2416                     @@FLAGS                  ;IY => flag table base
2416 3E65                LD      A,1Ø1
2418 EF                  RST     4Ø
2419 FDCBØ25E            BIT     3,(IY+'C'-'A')   ;System request?
241D CA3325              JP      Z,VIASET         ;"Install with Set
                 ;
                 ;       Grab system dependent vectors
                 ;
242Ø FDE5                PUSH    IY               ;Set DE to flag base
2422 D1                  POP     DE
2423 21ØAØØ              LD      HL,'K'-'A'       ;KFLAG$
2426 19                  ADD     HL,DE
2427 223127              LD      (KFLAG),HL       ;Save keyboard flag locn
242A 2112ØØ              LD      HL,'S'-'A'       ;SFLAG$
242D 19                  ADD     HL,DE
242E 225327              LD      (SFLAG),HL       ;Save system flag location
2431 2116ØØ              LD      HL,'W'-'A'       ;WRINT$
2434 19                  ADD     HL,DE
2435 229Ø26              LD      (WRINT),HL       ;Save int mask
2438 21DEFF              LD      HL,1Ø-44         ;INTVC$+1Ø
243B 19                  ADD     HL,DE            ;Save for receive int
243C 228D26              LD      (INTVC),HL       ;  vector
                 ;
                 ;       Move @ICNFG vector into driver
```

```
                ;
243F FD7E1C             LD      A,(IY+28)       ;Get current opcode
2442 329A26             LD      (LINK),A        ;Save in driver
2445 FD6E1D             LD      L,(IY+29)       ;Get current address
2448 FD661E             LD      H,(IY+30)       ;Put in driver code
244B 229B26             LD      (LINK+1),HL
                ;
                ;       Check if driver alread resident
                ;
244E 115725             LD      DE,CL$          ;Check if driver is
2451                    @@GTMOD                 ;  already resident
2451 3E53               LD      A,83
2453 EF                 RST     40
2454 EB                 EX      DE,HL           ;Put DCB ptr to HL
2455 201A               JR      NZ,NOTRES       ;Go if not
                ;
                ;       Make sure that the new DCB is same as the old
                ;
2457 4E                 LD      C,(HL)          ;P/u DCB pointer LSB
2458 23                 INC     HL
2459 46                 LD      B,(HL)          ;P/u DCB pointer MSB
245A 210600             LD      HL,6            ;Get old DCB name &
245D 09                 ADD     HL,BC           ;  stuff into error
245E 7E                 LD      A,(HL)          ;  message in case
245F 2C                 INC     L               ;  a different DCB
2460 66                 LD      H,(HL)          ;  is referenced
2461 6F                 LD      L,A
2462 221D26             LD      (DCBNAM$),HL    ;Stuff message with spec
2465 2A7F26             LD      HL,(CLDCB)      ;P/u DCB existing DCB
2468 B7                 OR      A               ;  pointer
2469 ED42               SBC     HL,BC           ;Same DCB pointer?
246B C23725             JP      NZ,DCBERR       ;Can't install if diff
246E C31425             JP      ISRES
2471 114B49       NOTRES LD     DE,'IK'
2474                    @@GTDCB                 ;Locate low memory ptr
2474 3E52               LD      A,82
2476 EF                 RST     40
2477 C24825             JP      NZ,IOERR        ;Go if not found
247A 2D                 DEC     L
247B 56                 LD      D,(HL)          ;P/u pointer to
247C 2D                 DEC     L               ;  start of free
247D 5E                 LD      E,(HL)          ;  low core
247E 22FA24             LD      (LCPTR+1),HL    ;Save ptr for later
2481 21EF00             LD      HL,CLEND-CLDVR-1
2484 19                 ADD     HL,DE           ;Start + driver length
2485 22C424             LD      (SVEND+1),HL
2488 010013             LD      BC,1300H        ;Max addr + 1
248B AF                 XOR     A
248C ED42               SBC     HL,BC           ;See if room low
248E 382D               JR      C,PUTLOW        ;  and install there if so
                ;
                ;       Check if high memory available
                ;
2490 FDCB0246           BIT     0,(IY+'C'-'A')  ;Memory frozen?
2494 C23B25             JP      NZ,NOROOM       ;Can't install if so
2497 210000             LD      HL,0
249A 45                 LD      B,L             ;Get HIGH$
249B                    @@HIGH$
249B 3E64               LD      A,100
249D EF                 RST     40
249E 22C424             LD      (SVEND+1),HL    ;Top of driver
```

```
24A1 B7                     OR       A
24A2 Ø1FØØØ                 LD       BC,CLEND-CLDVR   ;  minus length
24A5 ED42                   SBC      HL,BC
24A7 Ø6ØØ                   LD       B,Ø
24A9 E5                     PUSH     HL
24AA                        @@HIGH$                   ;  is new HIGH$
24AA 3E64                   LD       A,1ØØ
24AC EF                     RST      4Ø
24AD E1                     POP      HL
24AE 23                     INC      HL               ;Plus one is start
24AF 225525                 LD       (HCPTR),HL       ;Save it
24B2 215525                 LD       HL,HCPTR         ;  and point to it
24B5 22FA24                 LD       (LCPTR+1),HL
24B8 3EFF                   LD       A,ØFFH           ;Flag himem used
24BA 322525                 LD       (HGHFLG),A
                   ;
                   ;        Relocate internal references in driver
                   ;
24BD DDE5          PUTLOW   PUSH     IX
24BF DD216727               LD       IX,RELTAB        ;Point to relocation tbl
24C3 21ØØØØ        SVEND    LD       HL,$-$           ;Find distance to move
24C6 227926                 LD       (CLDVR+2),HL     ;Set last byte used
24C9 116627                 LD       DE,CLEND-1
24CC B7                     OR       A                ;Clear carry flag
24CD ED52                   SBC      HL,DE
24CF 44                     LD       B,H              ;Move to BC
24DØ 4D                     LD       C,L
24D1 3EØD                   LD       A,TABLEN         ;Get table length
24D3 DD6EØØ        RLOOP    LD       L,(IX)           ;Get address to change
24D6 DD66Ø1                 LD       H,(IX+1)
24D9 5E                     LD       E,(HL)           ;P/U address
24DA 23                     INC      HL
24DB 56                     LD       D,(HL)
24DC EB                     EX       DE,HL            ;Offset it
24DD Ø9                     ADD      HL,BC
24DE EB                     EX       DE,HL
24DF 72                     LD       (HL),D           ;Put it back
24EØ 2B                     DEC      HL
24E1 73                     LD       (HL),E
24E2 DD23                   INC      IX
24E4 DD23                   INC      IX
24E6 3D                     DEC      A
24E7 2ØEA                   JR       NZ,RLOOP         ;Loop till done
24E9 DDE1                   POP      IX               ;Restore DCB
                   ;
                   ;        Set up @ICNFG
                   ;
24EB 218926                 LD       HL,INIT          ;Get (relocated)
24EC               RXØ1     EQU      $-2
24EE FD751D                 LD       (IY+29),L        ;  init address & put
24F1 FD741E                 LD       (IY+3Ø),H        ;  into system ICNFG area
24F4 3EC3                   LD       A,ØC3H           ;Get JP instruction
24F6 FD771C                 LD       (IY+28),A        ;Turn on ICNFG
                   ;
                   ;        Move driver
                   ;
24F9 21ØØØØ        LCPTR    LD       HL,$-$           ;Low core or himem pointer
24FC 5E                     LD       E,(HL)
24FD 2C                     INC      L
24FE 56                     LD       D,(HL)
24FF D5                     PUSH     DE               ;Save start
```

```
25ØØ 217726                LD      HL,CLDVR
25Ø3 Ø1FØØØ                LD      BC,CLEND-CLDVR  ;Calc driver length
25Ø6 EDBØ                  LDIR                    ;Move into place
25Ø8 2AFA24                LD      HL,(LCPTR+1)    ;If driver went low,
25ØB 73                    LD      (HL),E          ;  need to update new
25ØC 2C                    INC     L               ;  driver zone pointer
25ØD 72                    LD      (HL),D
                  ;
                  ;        Initialize the driver
                  ;
25ØE F3                    DI
25ØF CD8926                CALL    INIT            ;Init the UART
251Ø              RX11     EQU     $-2
2512 FB                    EI
                  ;
2513 D1                    POP     DE              ;Pop filter start
                  ;
2514 212Ø26       ISRES    LD      HL,CLACT$       ;Advise COM/DVR installed
2517 DD36ØØØ7              LD      (IX),7          ;Init DCB type to "C/P/G"
251B DD73Ø1                LD      (IX+1),E        ;  & stuff the driver
251E DD72Ø2                LD      (IX+2),D        ;  address
2521                       @@LOGOT
                           IFEQ    ØØH,1
                           LD      HL,
                           ENDIF
2521 3EØC                  LD      A,12
2523 EF                    RST     4Ø
2524 3EØØ                  LD      A,$-$           ;Did it use high memory?
2525              HGHFLG   EQU     $-1
2526 B7                    OR      A               ;NZ if high
2527 28Ø6                  JR      Z,NTHGH
2529 215Ø26                LD      HL,HMEM$        ;"Driver in himem...
252C                       @@LOGOT
                           IFEQ    ØØH,1
                           LD      HL,
                           ENDIF
252C 3EØC                  LD      A,12
252E EF                    RST     4Ø
252F 21ØØØØ       NTHGH    LD      HL,Ø            ;Init on error code
2532 C9                    RET                     ;  and exit
                  ;
                  ;        Error exits
                  ;
2533 213B26       VIASET   LD      HL,VIASET$      ;"Install with Set
2536 DD                    DB      ØDDH
2537 21Ø126       DCBERR   LD      HL,DCBERR$      ;"Driver being used already
253A DD                    DB      ØDDH
253B 21E725       NOROOM   LD      HL,NOROOM$      ;"Memory frozen
253E                       @@LOGOT
                           IFEQ    ØØH,1
                           LD      HL,
                           ENDIF
253E 3EØC                  LD      A,12
254Ø EF                    RST     4Ø
2541 21FFFF                LD      HL,-1           ;Set abort code
2544                       @@CKBRKC                ;Clear any break
2544 3E6A                  LD      A,1Ø6
2546 EF                    RST     4Ø
2547 C9                    RET
                  ;
2548 6F           IOERR    LD      L,A             ;Error code to HL
```

```
2549 2600        02140         LD      H,0
254B F6C0        02150         OR      0C0H            ;Set short,return
254D 4F          02160         LD      C,A             ;Error to C
254E             02170         @@ERROR                 ;  for error dsply
254E 3E1A        00035         LD      A,26
2550 EF          00036         RST     40
2551             02180         @@CKBRKC                ;Clear any break
2551 3E6A        00037         LD      A,106
2553 EF          00038         RST     40
2554 C9          02190         RET
                 02200 ;
                 02210 ;       Messages & Data tables
                 02220 ;
2555 0000        02230 HCPTR   DW      0               ;Save start if going to HIGH$
2557 24          02240 CL$     DB      '$CL',3
     43 4C 03
255B 52          02250 HELLO$  DB      'RS-232 Driver'
     53 2D 32 33 32 20 44 72
     69 76 65 72
                 02260 ;
2568             02270 *GET    CLIENT:3
                 03950 ;CLIENTS/ASM - File to establish sign-on headers
                 03960 ;
2568 20          03970         DB      ' - 6.2.0 - Copyright 1982/83/84 by Logical'
     2D 20 36 2E 32 2E 30 20
     2D 20 43 6F 70 79 72 69
     67 68 74 20 31 39 38 32
     2F 38 33 2F 38 34 20 62
     79 20 4C 6F 67 69 63 61
     6C
2592 20          03980         DB      ' Systems, Inc.      ',10
     53 79 73 74 65 6D 73 2C
     20 49 6E 63 2E 20 20 20
     20 20 20 0A
                 03990 ;
25A7 41          04000         DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 20 52 69 67 68 74
     73 20 52 65 73 65 72 76
     65 64 2E 20 4C 69 63 65
     6E 73 65 64 20 31 39 38
     32 2F 38 33 2F 38 34
25CF 20          04010         DB      ' to xxxxxxxxxxxxxxxxxx',10,13
     74 6F 20 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 0A 0D
                 02280 ;
25E7 4E          02290 NOROOM$ DB      'No memory space available',CR
     6F 20 6D 65 6D 6F 72 79
     20 73 70 61 63 65 20 61
     76 61 69 6C 61 62 6C 65
     0D
2601 44          02300 DCBERR$ DB      'Driver already attached to *xx',CR
     72 69 76 65 72 20 61 6C
     72 65 61 64 79 20 61 74
     74 61 63 68 65 64 20 74
     6F 20 2A 78 78 0D
261D             02310 DCBNAM$ EQU     $-3
2620 43          02320 CLACT$  DB      'COM driver is now resident',CR
     4F 4D 20 64 72 69 76 65
     72 20 69 73 20 6E 6F 77
     20 72 65 73 69 64 65 6E
     74 ØD
263B 4D         Ø233Ø VIASET$ DB      'Must install via SET',CR
     75 73 74 2Ø 69 6E 73 74
     61 6C 6C 2Ø 76 69 61 2Ø
     53 45 54 ØD
265Ø ØA         Ø234Ø HMEM$   DB      LF,'Note: driver installed in high memory',CR
     4E 6F 74 65 3A 2Ø 64 72
     69 76 65 72 2Ø 69 6E 73
     74 61 6C 6C 65 64 2Ø 69
     6E 2Ø 68 69 67 68 2Ø 6D
     65 6D 6F 72 79 ØD
                Ø235Ø ;
ØØEØ            Ø236Ø @WRINT  EQU     ØEØH
ØØ8Ø            Ø237Ø WRINT$  EQU     8ØH
                Ø238Ø ;
ØØE8            Ø239Ø MASRES  EQU     ØE8H            ;RS232 ports
ØØE8            Ø24ØØ MODSTAT EQU     ØE8H
ØØE9            Ø241Ø BAUDSET EQU     ØE9H
ØØEA            Ø242Ø UARTCTL EQU     ØEAH
ØØEA            Ø243Ø UARTST  EQU     ØEAH
ØØEB            Ø244Ø DATAREG EQU     ØEBH
                Ø245Ø ;
                Ø246Ø ;       Actual driver
                Ø247Ø ;
2677            Ø248Ø CLDVR   EQU     $
2677 1831       Ø249Ø         JR      CLBGN           ;Branch around linkage
2679 6727       Ø25ØØ         DW      CLEND           ;Last byte used
267B Ø3         Ø251Ø         DB      3,'$CL'
     24 43 4C
267F ØØØØ       Ø252Ø CLDCB   DW      $-$
2681 ØØØØ       Ø253Ø         DW      Ø
2683            Ø254Ø CLDATA$ EQU     $
ØØØØ            Ø255Ø MSMASK  EQU     $-CLDATA$
2683 ØØ         Ø256Ø         DB      Ø
                Ø257Ø ;
                Ø258Ø ;       UART control port image
                Ø259Ø ;
                Ø26ØØ ;       bit 7: 1 = even parity, Ø = odd parity
                Ø261Ø ;       bits 6,5: word length <ØØ=5, 1Ø=6, Ø1=7, 11=8>
                Ø262Ø ;       bit 4: 1 = 2 stop bits, Ø = 1 stop bit
                Ø263Ø ;       bit 3: 1 = disable parity, Ø = enable parity
                Ø264Ø ;       bit 2: 1 = enable transmit data, Ø = break
                Ø265Ø ;       bit 1: Ø = Data Terminal Ready
                Ø266Ø ;       bit Ø: Ø = Request to Send
                Ø267Ø ;
ØØØ1            Ø268Ø UCIMAGE EQU     $-CLDATA$
2684 A5         Ø269Ø         DB      ØA5H
2685 55         Ø27ØØ BAUDRT  DB      55H             ;Init 3ØØ baud
ØØØ3            Ø271Ø LOGBRK  EQU     $-CLDATA$
2686 Ø3         Ø272Ø         DB      3               ;Default is Control-C
ØØØ4            Ø273Ø CLFLG   EQU     $-CLDATA$
2687 ØØ         Ø274Ø         DB      Ø               ;Init no char in buf
ØØØ5            Ø275Ø CLBUF   EQU     $-CLDATA$
2688 ØØ         Ø276Ø         DB      Ø               ;One-char buffer
                Ø277Ø ;
                Ø278Ø ;       CL initialization routine. Set up DR interrupt
                Ø279Ø ;       vector & initialize the hardware
                Ø28ØØ ;
2689 211827     Ø281Ø INIT    LD      HL,RECVINT      ;Vector address
268A            Ø282Ø RXØ2    EQU     $-2
268C 22ØØØØ     Ø283Ø         LD      ($-$),HL        ;INTVC$+1Ø
```

```
268D           Ø284Ø INTVC    EQU     $-2
268F 218ØØØ    Ø285Ø          LD      HL,WRINT$         ;Interrupt enable mask
269Ø           Ø286Ø WRINT    EQU     $-2
2692 CBEE      Ø287Ø          SET     5,(HL)            ;Enable RS232 DR
2694 7E        Ø288Ø          LD      A,(HL)
2695 D3EØ      Ø289Ø          OUT     (@WRINT),A
2697 CD9D26    Ø29ØØ          CALL    CTL2              ;Init the hardware
2698           Ø291Ø RXØ3     EQU     $-2
269A C9        Ø292Ø LINK     RET                       ;Link back thru any
269B ØØ        Ø293Ø          DB      Ø,Ø               ;  existing ICNFG
     ØØ
               Ø294Ø ;
               Ø295Ø ;        Initialize the UART & BRG
               Ø296Ø ;
269D ED4B8426  Ø297Ø CTL2     LD      BC,(CLDATA$+UCIMAGE)    ;P/u values from DCB
269F           Ø298Ø RXØ4     EQU     $-2
26A1 D3E8      Ø299Ø          OUT     (MASRES),A        ;Reprime UART
26A3 79        Ø3ØØØ          LD      A,C
26A4 D3EA      Ø3Ø1Ø          OUT     (UARTCTL),A
26A6 78        Ø3Ø2Ø          LD      A,B
26A7 D3E9      Ø3Ø3Ø          OUT     (BAUDSET),A
26A9 C9        Ø3Ø4Ø          RET
               Ø3Ø5Ø ;
26AA DD218326  Ø3Ø6Ø CLBGN    LD      IX,CLDATA$        ;Point to data area
26AC           Ø3Ø7Ø RXØ5     EQU     $-2
26AE 3855      Ø3Ø8Ø          JR      C,RECV            ;Go if @GET request
26BØ 2841      Ø3Ø9Ø          JR      Z,SEND            ;Go if @PUT request
26B2 79        Ø31ØØ          LD      A,C               ;P/U @CTL byte
26B3 B7        Ø311Ø          OR      A                 ;@CTL ØØ ?
26B4 2826      Ø312Ø          JR      Z,CANISND         ;Go if so
26B6 3D        Ø313Ø          DEC     A                 ;@CTL Ø1 ?
26B7 2857      Ø314Ø          JR      Z,CTL1            ;Go if so
26B9 3D        Ø315Ø          DEC     A                 ;Was it CTL-2 "INIT UART"
26BA 28E1      Ø316Ø          JR      Z,CTL2            ;Go if so
26BC FEØ2      Ø317Ø          CP      4-2               ;Wakeup feature?
26BE 28Ø2      Ø318Ø          JR      Z,CTL4            ;Go if wakeup feature
26CØ AF        Ø319Ø          XOR     A
26C1 C9        Ø32ØØ          RET
               Ø321Ø ;
26C2 FDE5      Ø322Ø CTL4     PUSH    IY                ;Transfer pointer to HL
26C4 E1        Ø323Ø          POP     HL
26C5 7C        Ø324Ø          LD      A,H               ;Test if set or reset
26C6 B5        Ø325Ø          OR      L
26C7 3EC9      Ø326Ø          LD      A,ØC9H            ;Init disable wakeup
26C9 EB        Ø327Ø          EX      DE,HL             ;Switch new value to DE
26CA 2A2127    Ø328Ø          LD      HL,(WAKEADR+1)    ;  & p/u old in HL
26CB           Ø329Ø RXØ6     EQU     $-2
26CD 28Ø2      Ø33ØØ          JR      Z,SETWAK          ;Jump if disable
26CF 3EC3      Ø331Ø          LD      A,ØC3H            ;Make enable
26D1 322Ø27    Ø332Ø SETWAK   LD      (WAKEADR),A       ;Load the opcode
26D2           Ø333Ø RXØ7     EQU     $-2
26D4 ED532127  Ø334Ø          LD      (WAKEADR+1),DE    ;Then the address
26D6           Ø335Ø RXØ8     EQU     $-2
26D8 E5        Ø336Ø          PUSH    HL                ;Transfer pointer to IY
26D9 FDE1      Ø337Ø          POP     IY
26DB C9        Ø338Ø          RET
               Ø339Ø ;
               Ø34ØØ ;        Check if ready to send
               Ø341Ø ;
26DC DBEA      Ø342Ø CANISND  IN      A,(UARTST)        ;Look at TX empty bit
26DE 2F        Ø343Ø          CPL                       ;Flip it
```

```
26DF E64Ø      Ø344Ø          AND     4ØH              ;Mask out all else
26E1 DBE8      Ø345Ø          IN      A,(MODSTAT)      ;P/U modem status reg
26E3 CØ        Ø346Ø          RET     NZ               ;Return if can't send
26E4 47        Ø347Ø          LD      B,A              ;Save modem status reg
26E5 DDAEØØ    Ø348Ø          XOR     (IX+MSMASK)      ;Mask for which to flip
26E8 1F        Ø349Ø          RRA                      ;Move into bits Ø-3
26E9 1F        Ø35ØØ          RRA
26EA 1F        Ø351Ø          RRA
26EB 1F        Ø352Ø          RRA
26EC DDA6ØØ    Ø353Ø          AND     (IX+MSMASK)      ;Mask for which to check
26EF E6ØF      Ø354Ø          AND     ØFH              ;Mask off garbage
26F1 78        Ø355Ø          LD      A,B              ;Get back reg
26F2 C9        Ø356Ø          RET                      ;Ret with Z or NZ
               Ø357Ø ;
               Ø358Ø ;        Send character
               Ø359Ø ;
26F3 DD7EØ1    Ø36ØØ SEND     LD      A,(IX+UCIMAGE)   ;Get UART ctrl reg
26F6 D3EA      Ø361Ø          OUT     (UARTCTL),A      ;Put it (clears BREAK)
26F8 CDDC26    Ø362Ø SWAIT    CALL    CANISND          ;Poll
26F9           Ø363Ø RXØ9     EQU     $-2
26FB 2ØFB      Ø364Ø          JR      NZ,SWAIT         ; until ready
26FD 79        Ø365Ø          LD      A,C              ;Get byte to send
26FE D3EB      Ø366Ø          OUT     (DATAREG),A      ;Send it with Z-flag
27ØØ C9        Ø367Ø          RET                      ;  unchanged for return
               Ø368Ø ;
               Ø369Ø ;        Receive character - Get from buffer if available
               Ø37ØØ ;
27Ø1 CD2327    Ø371Ø RECV1    CALL    CKINP            ;Ck if avail from port
27Ø2           Ø372Ø RX1Ø     EQU     $-2
27Ø4 CØ        Ø373Ø          RET     NZ               ;Back if none
27Ø5 DDCBØ426  Ø374Ø RECV     SLA     (IX+CLFLG)       ;Ck if avail from buf
27Ø9 3ØF6      Ø375Ø          JR      NC,RECV1         ;Go if none avail
27ØB DD7EØ5    Ø376Ø          LD      A,(IX+CLBUF)     ;Get the char
27ØE BF        Ø377Ø          CP      A                ;Set Z-flag & exit
27ØF C9        Ø378Ø          RET
               Ø379Ø ;
               Ø38ØØ ;        Break request
               Ø381Ø ;
271Ø DD7EØ1    Ø382Ø CTL1     LD      A,(IX+UCIMAGE)   ;Pick up UART ctl image
2713 CB97      Ø383Ø          RES     2,A              ;Show BREAK bit
2715 D3EA      Ø384Ø          OUT     (UARTCTL),A
2717 C9        Ø385Ø          RET                      ;With Z-flag
               Ø386Ø ;
               Ø387Ø ;        Data received interrupt handler
               Ø388Ø ;
2718 DD218326  Ø389Ø RECVINT  LD      IX,CLDATA$       ;Base of data area
271A           Ø39ØØ RX13     EQU     $-2
271C CD2327    Ø391Ø          CALL    CKINP            ;See if available from port
271D           Ø392Ø RX12     EQU     $-2
271F 78        Ø393Ø          LD      A,B
272Ø C9        Ø394Ø WAKEADR  RET                      ;Wakeup if enabled
2721 ØØØØ      Ø395Ø          DW      Ø                ;Space for address
               Ø396Ø ;
               Ø397Ø ;        Routine to check on a received character
               Ø398Ø ;
2723 DBEA      Ø399Ø CKINP    IN      A,(UARTST)       ;Check if actually RX
2725 47        Ø4ØØØ          LD      B,A              ;Save status
2726 E68Ø      Ø4Ø1Ø          AND     8ØH              ;Mask Data Received bit
2728 EE8Ø      Ø4Ø2Ø          XOR     8ØH              ;Set NZ if none avail
272A 3EØØ      Ø4Ø3Ø          LD      A,Ø              ;Set "No error"
272C CØ        Ø4Ø4Ø          RET     NZ               ;Return if none
```

```
272D DBEB       04050          IN      A,(DATAREG)     ;Pick up character
272F 4F         04060          LD      C,A             ;Save tempy in reg-C
                04070 ;
                04080 ;        Break, Pause & Enter handler routine
                04090 ;
2730 210000     04100          LD      HL,$-$          ;KFLAG$
2731            04110 KFLAG    EQU     $-2
2733 FE0D       04120          CP      CR              ;ENTER char received?
2735 2004       04130          JR      NZ,PAWSCK       ;Go if not
2737 CBD6       04140          SET     2,(HL)          ;Set ENTER bit
2739 1823       04150          JR      RECVEX
                04160 ;
273B FE60       04170 PAWSCK   CP      60H             ;Pause char received?
273D 2004       04180          JR      NZ,BRKCHK       ;Go if not
273F CBCE       04190          SET     1,(HL)          ;Set pause bit
2741 181B       04200          JR      RECVEX
                04210 ;
2743 DD7E03     04220 BRKCHK   LD      A,(IX+LOGBRK)   ;Break char received?
2746 B7         04230          OR      A               ;Check if LOGBRK=0
2747 2815       04240          JR      Z,RECVEX        ;No valid break if =0
2749 B9         04250          CP      C               ;Check if a valid BREAK
274A 2806       04260          JR      Z,BRKRECD       ;Go if so
274C DBEA       04270          IN      A,(UARTST)      ;Check for framing error
274E CB67       04280          BIT     4,A
2750 280C       04290          JR      Z,RECVEX        ;Quit if none
                04300 ;
                04310 ;        A BREAK was received, ck system's BREAK disable
                04320 ;
2752 3A0000     04330 BRKRECD  LD      A,($-$)         ;Check if break key
2753            04340 SFLAG    EQU     $-2
2755 E610       04350          AND     10H             ;  is disabled
2757 3E00       04360          LD      A,0             ;Return NZ & A=0 if
2759 C0         04370          RET     NZ              ;  the BREAK is disabled
275A CBC6       04380          SET     0,(HL)          ;Else set break bit
275C 0E80       04390          LD      C,80H           ;  & reset BREAK code
275E DD7105     04400 RECVEX   LD      (IX+CLBUF),C    ;Put char into 1-char buf
2761 DD360480   04410          LD      (IX+CLFLG),80H  ;  & set char available
2765 AF         04420          XOR     A               ;Set Z flag
2766 C9         04430          RET
2767            04440 CLEND    EQU     $
                04450 ;
2767 EC24       04460 RELTAB   DW      RX01,RX02,RX03,RX04,RX05,RX06,RX07,RX08
     8A26 9826 9F26 AC26 CB26 D226 D626
2777 F926       04470          DW      RX09,RX10,RX11,RX12,RX13
     0227 1025 1D27 1A27
000D            04480 TABLEN   EQU     $-RELTAB/2
                04490 ;
2400            04500          END     BEGIN
```

```
@@1               ØØØØ @@2               ØØØØ @@3               ØØØØ
@@4               ØØØØ @MOD2             ØØØØ @MOD4             FFFF
@WRINT            ØØEØ BAUDRT            2685 BAUDSET           ØØE9
BEGIN             24ØØ BEGINA            24Ø9 BRKCHK            2743
BRKRECD           2752 CANISND           26DC CKINP             2723
CL$               2557 CLACT$            262Ø CLBGN             26AA
CLBUF             ØØØ5 CLDATA$           2683 CLDCB             267F
CLDVR             2677 CLEND             2767 CLFLG             ØØØ4
CR                ØØØD CTL1              271Ø CTL2              269D
CTL4              26C2 DATAREG           ØØEB DCBERR            2537
DCBERR$           26Ø1 DCBNAM$           261D HCPTR             2555
HELLO$            255B HGHFLG            2525 HMEM$             265Ø
INIT              2689 INTVC             268D IOERR             2548
ISRES             2514 KFLAG             2731 LCPTR             24F9
LF                ØØØA LINK              269A LOGBRK            ØØØ3
MASRES            ØØE8 MODSTAT           ØØE8 MSMASK            ØØØØ
NOROOM            253B NOROOM$           25E7 NOTRES            2471
NTHGH             252F PAWSCK            273B PUTLOW            24BD
RECV              27Ø5 RECV1             27Ø1 RECVEX            275E
RECVINT           2718 RELTAB            2767 RLOOP             24D3
RXØ1              24EC RXØ2              268A RXØ3              2698
RXØ4              269F RXØ5              26AC RXØ6              26CB
RXØ7              26D2 RXØ8              26D6 RXØ9              26F9
RX1Ø              27Ø2 RX11              251Ø RX12              271D
RX13              271A SEND              26F3 SETWAK            26D1
SFLAG             2753 SVEND             24C3 SWAIT             26F8
TABLEN            ØØØD UARTCTL           ØØEA UARTST            ØØEA
UCIMAGE           ØØØ1 VIASET            2533 VIASET$           263B
WAKEADR           272Ø WRINT             269Ø WRINT$            ØØ8Ø
@@ABORT           948A @@ADTSK           951D @@BANK            9A35
@@BKSP            9715 @@BREAK           9A4B @@CHNIO           9475
@@CKBRKC          9A99 @@CKDRV           9571 @@CKEOF           972A
@@CKTSK           95Ø8 @@CLOSE           97ØØ @@CLS             9A83
@@CMNDI           94B4 @@CMNDR           94C9 @@CTL             92D9
@@DATE            944B @@DCSTAT          95BØ @@DEBUG           94F3
@@DECHEX          99B5 @@DIRRD           9922 @@DIRWR           9937
@@DIV16           99AØ @@DIV8            998B @@DODIR           9586
@@DSP             929D @@DSPLY           933D @@ERROR           94DE
@@EXIT            949F @@FEXT            988F @@FLAGS           9A1F
@@FNAME           98A4 @@FSPEC           987A @@GATRD           99ØD
@@GATWR           994C @@GET             92B1 @@GTDCB           98CE
@@GTDCT           98B9 @@GTMOD           98E3 @@HDFMT           9658
@@HEX16           99F4 @@HEX8            99DF @@HEXDEC          99CA
@@HIGH$           9AØ9 @@INIT            96D6 @@KBD             9315
@@KEY             9289 @@KEYIN           9329 @@KLTSK           955C
@@LOAD            985Ø @@LOC             973F @@LOF             9754
@@LOGER           9374 @@LOGOT           9389 @@MSG             93CØ
@@MUL16           9976 @@MUL8            9961 @@OPEN            96EB
@@PARAM           9436 @@PAUSE           9421 @@PEOF            9769
@@POSN            977E @@PRINT           93D5 @@PRT             92ED
@@PUT             92C5 @@RAMDIR          959B @@RDSEC           962E
@@RDSSC           98F8 @@READ            9793 @@REMOV           96C1
@@RENAM           96AC @@REW             97A8 @@RMTSK           9532
@@RPTSK           9547 @@RREAD           97BD @@RSLCT           9619
@@RSTOR           95DA @@RUN             9865 @@RWRIT           97D2
@@SEEK            96Ø4 @@SEEKSC          97E7 @@SKIP            97FC
@@SLCT            95C5 @@STEPI           95EF @@TIME            946Ø
@@VDCTL           94ØC @@VER             9811 @@VRSEC           9643
@@WEOF            9826 @@WHERE           93Ø1 @@WRITE           983B
@@WRSEC           966D @@WRSSC           9682 @@WRTRK           9697
```

```
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

COMM/CMD - Terminal program with file send and receive

The Comm utility program acts as a terminal for communications work. Its features include file send and receive, and fully buffered device I/O (including printer spooling).

```
0000           *GET    LCOMM
               ;LCOMM/ASM - COMM Communications Program
0000                   TITLE   <COMM - LS-DOS 6.2>
0000                   SUBTTL  '<Program Code Section>'
               ;
FFFF           BUFFRD  EQU     -1                  ;Set true
0080           BREAK   EQU     80H                 ;Char fm keyboard
000A           LF      EQU     10
000D           CR      EQU     13
0013           XOFF    EQU     'S'&1FH
               ;
0000           *GET    SVCMAC:3                    ;SVC Macro equivalents
               ;SVCMAC/ASM - LS-DOS Version VI
               *LIST   OFF
               *LIST   ON
0000           *GET    COPYCOM:3                   ;Copyright messages
               ; COPYCOM - File for Copyright COMment block
               ;
0000                   COM     '<*(C) 1982,83,84 by LSI*>'
               ;
3000           BASE    EQU     3000H
3000                   ORG     BASE
               ;
3000 210000    $EXIT   LD      HL,0                ;Init no error
3003 310000    QUIT$   LD      SP,$-$              ;P/u original stack
3004           STACK   EQU     $-2
3006                   @@CKBRKC                    ;Clear break bit
3005 3E6A              LD      A,106
3008 EF                RST     40
3009 C9                RET
               ;
300A 21FFFF    $ABORT  LD      HL,-1               ;Set abort code
300D 18F4              JR      QUIT$
               ;
300F E5        $OPEN   PUSH    HL
3010 210000            LD      HL,$-$              ;Address of SFLAG$
3011           SFLG    EQU     $-2
3013 CBC6              SET     0,(HL)              ;Set open inhibit bit
3015 E1                POP     HL
3016                   @@OPEN                      ;Do the open
3016 3E3B              LD      A,59
3018 EF                RST     40
3019 C9                RET                         ;Return with status
               ;
301A C5        $ERROR  PUSH    BC
301B F6C0              OR      0C0H                ;Set short,return
301D 4F                LD      C,A                 ;Error code to C
301E                   @@ERROR                     ;  for error display
301E 3E1A              LD      A,26
3020 EF                RST     40
3021 C1                POP     BC
3022 C9                RET
               ;
3023 3E00      MAINLP  LD      A,0                 ;Test warning flag set
3025 B7                OR      A                   ;  by OUTPUT on NEXTAP
3026 280F              JR      Z,ENUFPG            ;Go if > 2K of space
3028 21AC37            LD      HL,LILPG$           ;Display warning
302B                   @@DSPLY
                       IFEQ    00H,1
                       LD      HL,
                       ENDIF
```

Program Code Section

```
3Ø2B 3EØA      ØØØ1Ø          LD      A,1Ø
3Ø2D EF        ØØØ11          RST     4Ø
3Ø2E 3E13      Ø44ØØ          LD      A,XOFF          ;Schedule a forced PUT
3Ø2F           Ø441Ø XOFFP2   EQU     $-1
3Ø3Ø 325234    Ø442Ø          LD      (FRCPUT+1),A
3Ø33 AF        Ø443Ø          XOR     A
3Ø34 32243Ø    Ø444Ø          LD      (MAINLP+1),A    ;Inhibit until next page
3Ø37 DD219538  Ø445Ø ENUFPG   LD      IX,KIVCTR       ;Get key from buffer if
3Ø3B CDB438    Ø446Ø          CALL    PGMGET          ;  available
3Ø3E 2Ø22      Ø447Ø          JR      NZ,SENDIT       ;Bypass if got one
3Ø4Ø 3EØØ      Ø448Ø FSSW     LD      A,Ø             ;FS On/Off (XMIT File)
3Ø42 B7        Ø449Ø          OR      A
3Ø43 2832      Ø45ØØ          JR      Z,FSOFF         ;Bypass if not XMTG
3Ø45 3AAD38    Ø451Ø CKFREPG  LD      A,(FREEPG)      ;Don't get from file
3Ø48 FEØC      Ø452Ø          CP      12              ;  if < 3K buffer space
3Ø4A DA773Ø    Ø453Ø          JP      C,FSOFF         ;Go if less
3Ø4D 11Ø438    Ø454Ø          LD      DE,FS_FCB       ;Get sending FCB
3Ø5Ø           Ø455Ø FSSWGO   @@GET                   ;Get a byte to XMIT
3Ø5Ø 3EØ3      ØØØ12          LD      A,3
3Ø52 EF        ØØØ13          RST     4Ø
3Ø53 28ØD      Ø456Ø          JR      Z,SENDIT        ;Bypass if got byte
3Ø55 FE1C      Ø457Ø          CP      1CH             ;EOF encountered?
3Ø57 28Ø3      Ø458Ø          JR      Z,EOFFS         ;Bypass if EOF
3Ø59 CD1A3Ø    Ø459Ø          CALL    $ERROR          ;Output error message
3Ø5C CDDA33    Ø46ØØ EOFFS    CALL    FS_OFF          ;Turn off XMIT
3Ø5F C3F43Ø    Ø461Ø          JP      SKIPREC         ;  and ignore this round
3Ø62 4F        Ø462Ø SENDIT   LD      C,A             ;Xfer byte
3Ø63 FEØØ      Ø463Ø XLTS1    CP      Ø               ;Single character send
3Ø65 2ØØ2      Ø464Ø          JR      NZ,DPLXSW       ;  translate table
3Ø67 ØEØØ      Ø465Ø XLTS2    LD      C,Ø
3Ø69 Ø6ØØ      Ø466Ø DPLXSW   LD      B,Ø             ;Duplex On/Off
3Ø6B Ø4        Ø467Ø          INC     B
3Ø6C Ø5        Ø468Ø          DEC     B               ;Display on our devices
3Ø6D C41631    Ø469Ø          CALL    NZ,DEVOUT       ;  if duplex on (half)
3Ø7Ø 3AFA33    Ø47ØØ LCMON    LD      A,(TASK8A+2)    ;Ck CL on
3Ø73 B7        Ø471Ø          OR      A
3Ø74 C4Ø231    Ø472Ø          CALL    NZ,SNDOUT       ;Send char if ON
3Ø77 3AFA33    Ø473Ø FSOFF    LD      A,(TASK8A+2)    ;Test for CL ON
3Ø7A B7        Ø474Ø          OR      A
3Ø7B CAF43Ø    Ø475Ø          JP      Z,SKIPREC       ;Go if not
3Ø7E DD219D38  Ø476Ø          LD      IX,CLREC
3Ø82 CDB438    Ø477Ø          CALL    PGMGET          ;Ck for char avail
3Ø85 CAF43Ø    Ø478Ø          JP      Z,SKIPREC       ;Go if no char
3Ø88 Ø6ØØ      Ø479Ø DSPCTRL  LD      B,Ø             ;Ck if display of control
3Ø8A Ø4        Ø48ØØ          INC     B               ;  codes is in effect
3Ø8B Ø5        Ø481Ø          DEC     B
3Ø8C 2813      Ø482Ø          JR      Z,SAVCHR        ;Go if no ctrl display
3Ø8E FE2Ø      Ø483Ø          CP      2ØH
3Ø9Ø 3ØØF      Ø484Ø          JR      NC,SAVCHR       ;Go if not ctrl
3Ø92 F5        Ø485Ø          PUSH    AF              ;Save the char
3Ø93 21BF35    Ø486Ø          LD      HL,BRAKET+1     ;Pt to control char msg
3Ø96 4F        Ø487Ø          LD      C,A
3Ø97           Ø488Ø          @@HEX8                  ;Cvrt char & stuff in buf
3Ø97 3E62      ØØØ14          LD      A,98
3Ø99 EF        ØØØ15          RST     4Ø
3Ø9A 21BE35    Ø489Ø          LD      HL,BRAKET       ;Start of msg string
3Ø9D           Ø49ØØ          @@DSPLY                 ;Display ASCII control value
               ØØØ16          IFEQ    ØØH,1
               ØØØ17          LD      HL,
```

Program Code Section

```
                         ENDIF
3Ø9D 3EØA                LD      A,1Ø
3Ø9F EF                  RST     4Ø
3ØAØ F1                  POP     AF               ;Rcvr char
3ØA1 4F         SAVCHR   LD      C,A              ;Save char
3ØA2 Ø6ØØ       SHAKE    LD      B,Ø              ;Handshake On/Off
3ØA4 Ø4                  INC     B
3ØA5 Ø5                  DEC     B
3ØA6 282Ø                JR      Z,ECHOSW         ;Go if off
3ØA8 FE11                CP      'Q'&1FH          ;Ctrl-Q?
3ØA9            XONP1    EQU     $-1              ;Modify if PARM
3ØAA 28Ø6                JR      Z,CTLQ           ;Go if so
3ØAC FE13                CP      'S'&1FH          ;Ctrl-S?
3ØAD            XOFFP1   EQU     $-1              ;Modify if parm entered
3ØAE 2ØØ8                JR      NZ,NOSQ          ;Go if neither
3ØBØ Ø6ØØ                LD      B,Ø              ;Turn off
3ØB2 78         CTLQ     LD      A,B              ;  or on
3ØB3 324534              LD      (TASK8B+1),A     ;*CL send task
3ØB6 183C                JR      SKIPREC          ;Discard ctrl code
3ØB8 FE12       NOSQ     CP      'R'&1FH          ;Ctrl-R?
3ØBA 28Ø6                JR      Z,CTLR           ;Go if so
3ØBC FE14                CP      'T'&1FH          ;Ctrl-T?
3ØBE 2ØØ8                JR      NZ,ECHOSW        ;Go if neither
3ØCØ Ø6ØØ                LD      B,Ø              ;Turn off
3ØC2 78         CTLR     LD      A,B              ;  or on
3ØC3 322E31              LD      (FRSW+1),A       ;FR device
3ØC6 182C                JR      SKIPREC          ;Discard ctrl code
                ;
                ;        Test for ECHO after checking for handshake chars
                ;
3ØC8 Ø6ØØ       ECHOSW   LD      B,Ø              ;Echo On/Off?
3ØCA Ø4                  INC     B
3ØCB Ø5                  DEC     B
3ØCC C4FA3Ø              CALL    NZ,CLOUT         ;Send char back if ON
3ØCF 79                  LD      A,C
3ØDØ FEØD                CP      CR               ;Was it a CR?
3ØD2 2ØØB                JR      NZ,NOTCR
3ØD4 CDØ931              CALL    ECLF1            ;Send LF back if needed
3ØD7 21E13Ø              LD      HL,CRSW+1        ;Flag for CR recvd
                ;
                ;        Move state of ACCEPT LF switch into CRSW+1 when CR recv'd
                ;
3ØDA 3EØØ       ACCLFSW  LD      A,Ø              ;Show CR found if accept
3ØDC 77                  LD      (HL),A           ;  LF switch is off
3ØDD 1812                JR      TAKEREC          ;Dsp CR
                ;
                ;        When LF rcv'd, delete if ACCLFSW is off & last char was CR
                ;
3ØDF 79         NOTCR    LD      A,C              ;Check char
3ØEØ Ø6FF       CRSW     LD      B,ØFFH           ;P/u del LF switch
3ØE2 21E13Ø              LD      HL,CRSW+1        ;Pt to switch
3ØE5 36FF                LD      (HL),ØFFH        ; (flip off switch -not CR)
3ØE7 FEØA                CP      LF               ;Is line feed the char?
3ØE9 2ØØ6                JR      NZ,TAKEREC       ;Go if not LF
3ØEB 3A1F34              LD      A,(EIGHT+1)      ;Also skip if 8 bit
3ØEE BØ                  OR      B                ;  switch is off
3ØEF 28Ø3                JR      Z,SKIPREC        ;Skip LF if so
                ;
3ØF1 CD1631     TAKEREC  CALL    DEVOUT           ;Out to active devices
```

Program Code Section

```
3ØF4 CDE933     Ø547Ø SKIPREC CALL    TASKS           ;Do 3 tasks (incl kbd)
3ØF7 C3233Ø     Ø548Ø         JP      MAINLP          ;  & FRIO test then loop
                Ø549Ø ;
3ØFA 79         Ø55ØØ CLOUT   LD      A,C             ;Get char
3ØFB DD21A138   Ø551Ø         LD      IX,CLSEND       ;Set buffer pointers
3ØFF C3AE38     Ø552Ø         JP      OUTPGM          ;Put in output buffer
                Ø553Ø ;
31Ø2 CDFA3Ø     Ø554Ø SNDOUT  CALL    CLOUT           ;Send this character
31Ø5 79         Ø555Ø         LD      A,C             ;Is it CR?
31Ø6 FEØD       Ø556Ø         CP      CR
31Ø8 CØ         Ø557Ø         RET     NZ              ;Done if not
                Ø558Ø ;
31Ø9 3EØØ       Ø559Ø ECLF1   LD      A,$-$           ;Is echo linefeed on?
31ØA            Ø56ØØ ECOLF   EQU     $-1
31ØB B7         Ø561Ø         OR      A
31ØC C8         Ø562Ø         RET     Z               ;Done if not
31ØD 3EØA       Ø563Ø         LD      A,LF            ;Otherwise load a LF
31ØF DD21A138   Ø564Ø         LD      IX,CLSEND
3113 C3AE38     Ø565Ø         JP      OUTPGM          ;Add to buffer/ret to caller
                Ø566Ø ;
                Ø567Ø ;       Output to video
                Ø568Ø ;
3116 3EFF       Ø569Ø DEVOUT  LD      A,ØFFH          ;Is *DO On/Off?
3118 B7         Ø57ØØ         OR      A
3119 2812       Ø571Ø         JR      Z,FRSW          ;Bypass if off
311B 79         Ø572Ø         LD      A,C
311C FEØC       Ø573Ø         CP      ØCH             ;If formfeed,
311E 4F         Ø574Ø         LD      C,A
311F C5         Ø575Ø         PUSH    BC
312Ø 2ØØ7       Ø576Ø         JR      NZ,NOTCLS       ;  clear the screen
3122 ØE1C       Ø577Ø         LD      C,1CH           ;Cursor home
3124            Ø578Ø         @@DSP
3124 3EØ2       ØØØ21         LD      A,2
3126 EF         ØØØ22         RST     4Ø
3127 ØE1F       Ø579Ø         LD      C,1FH           ;Clear to end-of-frame
3129            Ø58ØØ NOTCLS  @@DSP
3129 3EØ2       ØØØ23         LD      A,2
312B EF         ØØØ24         RST     4Ø
312C C1         Ø581Ø         POP     BC
                Ø582Ø ;
                Ø583Ø ;       Send char to our disk if FR on
                Ø584Ø ;
312D 3EØØ       Ø585Ø FRSW    LD      A,Ø             ;FR On/Off - receive file
312F B7         Ø586Ø         OR      A
313Ø 28Ø8       Ø587Ø         JR      Z,PUTPR         ;Bypass if FR off
3132 79         Ø588Ø         LD      A,C
3133 DD21A938   Ø589Ø         LD      IX,FRVCTR       ;Put away into the
3137 CDAE38     Ø59ØØ         CALL    OUTPGM          ;  FR buffer
                Ø591Ø ;
                Ø592Ø ;       Place char into printer buffer if PR on
                Ø593Ø ;
313A 3EØØ       Ø594Ø PUTPR   LD      A,Ø             ;PR On/Off?
313C B7         Ø595Ø         OR      A
313D 28Ø8       Ø596Ø         JR      Z,FRIOSW        ;Go if off
313F 79         Ø597Ø         LD      A,C
314Ø DD219938   Ø598Ø         LD      IX,PRVCTR       ;Place the char in
3144 CDAE38     Ø599Ø         CALL    OUTPGM          ;  the printer buffer
                Ø6ØØØ ;
                Ø6Ø1Ø ;       Check if FR to disk is engaged
```

Program Code Section

```
                ØØ6Ø2Ø ;
3147 3EFF       Ø6Ø3Ø FRIOSW  LD      A,-1            ;Ck if FR-to-disk is on
3149 B7         Ø6Ø4Ø         OR      A
314A C8         Ø6Ø5Ø         RET     Z               ;Go if not engaged
314B DD21A938   Ø6Ø6Ø         LD      IX,FRVCTR       ;Is a char available
314F CDB438     Ø6Ø7Ø         CALL    PGMGET          ;  for the disk?
3152 C8         Ø6Ø8Ø         RET     Z               ;Go if none for disk
3153 212438     Ø6Ø9Ø         LD      HL,FR_FCB       ;Put char to disk
3156 CB7E       Ø61ØØ         BIT     7,(HL)          ;OPEN FCB?
3158 C8         Ø611Ø         RET     Z               ;Skip if not
3159 EB         Ø612Ø         EX      DE,HL
315A 4F         Ø613Ø         LD      C,A             ;Place char in "C"
315B            Ø614Ø         @@PUT                   ;  and do the write
315B 3EØ4       ØØØ25         LD      A,4
315D EF         ØØØ26         RST     4Ø
315E C8         Ø615Ø         RET     Z               ;Back if good
315F CD1A3Ø     Ø616Ø         CALL    $ERROR
3162 CDE433     Ø617Ø         CALL    FRIO_OFF        ;Turn FRIO to disk off
3165 C3DF33     Ø618Ø         JP      FR_OFF          ;Turn FR off and return
                Ø619Ø ;
                Ø62ØØ ;       <CLEAR> command function entered - decode it
                Ø621Ø ;
3168 Ø1ØØØØ     Ø622Ø CMDKEY  LD      BC,Ø            ;Init no device vector
316B 11ØØØØ     Ø623Ø         LD      DE,Ø            ;Init no File FCB
316E 21893Ø     Ø624Ø         LD      HL,DSPCTRL+1    ;Pt to ctrl char dsply parm
                Ø625Ø         IF      @MOD4
3171 FEA7       Ø626Ø         CP      27H!8ØH         ;Display control chars?
                Ø627Ø         ENDIF
                Ø628Ø         IF      @MOD2
                Ø629Ø         CP      '&'+8ØH
                Ø63ØØ         ENDIF
3173 CA3332     Ø631Ø         JP      Z,QFUNC
                Ø632Ø ;
3176 216A3Ø     Ø633Ø         LD      HL,DPLXSW+1
3179 FEA1       Ø634Ø         CP      '!'+8ØH         ;Ck duplex
317B CA3332     Ø635Ø         JP      Z,QFUNC
                Ø636Ø ;
317E 21C93Ø     Ø637Ø         LD      HL,ECHOSW+1
                Ø638Ø         IF      @MOD4
3181 FEA2       Ø639Ø         CP      '"'+8ØH         ;Ck echo
                Ø64ØØ         ENDIF
                Ø641Ø         IF      @MOD2
                Ø642Ø         CP      '@'+8ØH
                Ø643Ø         ENDIF
3183 CA3332     Ø644Ø         JP      Z,QFUNC
                Ø645Ø ;
3186 21A33Ø     Ø646Ø         LD      HL,SHAKE+1      ;Check handshake
                Ø647Ø         IF      @MOD4
3189 FEAA       Ø648Ø         CP      '*'+8ØH
                Ø649Ø         ENDIF
                Ø65ØØ         IF      @MOD2
                Ø651Ø         CP      '_'+8ØH
                Ø652Ø         ENDIF
318B CA4232     Ø653Ø         JP      Z,QSHAKE
                Ø654Ø ;
318E 21ØA31     Ø655Ø         LD      HL,ECOLF
3191 FEA3       Ø656Ø         CP      '#'+8ØH         ;Echo line feed?
3193 CA3332     Ø657Ø         JP      Z,QFUNC
                Ø658Ø ;
```

Program Code Section

```
3196 21DB30                LD      HL,ACCLFSW+1      ;Check accept-LF
3199 FEA4                  CP      '$'+80H
319B CA3332                JP      Z,QFUNC
                   ;
319E 211F34                LD      HL,EIGHT+1        ;Check 8-bit
                           IF      @MOD4
31A1 FEA9                  CP      ')'+80H
                           ENDIF
                           IF      @MOD2
                           CP      '('+80H
                           ENDIF
31A3 CA3332                JP      Z,QFUNC
                   ;
31A6 019538                LD      BC,KIVCTR         ;Init *KI put/get index
31A9 218134                LD      HL,KISW+1
31AC FEB1                  CP      '1'+80H           ;CK *KI
31AE CA3332                JP      Z,QFUNC
                   ;
31B1 010000                LD      BC,0              ;No *DO put/get index
31B4 211731                LD      HL,DEVOUT+1
31B7 FEB2                  CP      '2'+80H           ;CK *DO
31B9 2878                  JR      Z,QFUNC
                   ;
31BB 019938                LD      BC,PRVCTR         ;Init *PR put/get index
31BE 213B31                LD      HL,PUTPR+1
31C1 FEB3                  CP      '3'+80H           ;CK *PR
31C3 286E                  JR      Z,QFUNC
                   ;
31C5 01A138                LD      BC,CLSEND         ;Init *CL-S put/get index
31C8 21FA33                LD      HL,TASK8A+2
31CB FEB4                  CP      '4'+80H           ;CK *CL
31CD 2869                  JR      Z,QCL
                   ;
31CF 01A538                LD      BC,FSVCTR         ;Init *FS put/get index
31D2 110438                LD      DE,FS_FCB         ;Init *FS FCB
31D5 DD21003A              LD      IX,XMTBUF         ;Point to buffer
31D9 214130                LD      HL,FSSW+1
31DC FEB5                  CP      '5'+80H           ;CK FS
31DE 2853                  JR      Z,QFUNC
                   ;
31E0 01A938                LD      BC,FRVCTR         ;P/u *FR put/get index
31E3 112438                LD      DE,FR_FCB         ;P/u *FR FCB
31E6 DD21003B              LD      IX,RCVBUF         ;Pt to buffer
31EA 212E31                LD      HL,FRSW+1
31ED FEB6                  CP      '6'+80H           ;CK FR
31EF 2842                  JR      Z,QFUNC
                   ;
31F1 214831                LD      HL,FRIOSW+1
31F4 110000                LD      DE,0              ;No FCB here
31F7 FEB7                  CP      '7'!80H           ;Check FR IO to disk?
31F9 2838                  JR      Z,QFUNC
                   ;
31FB FEB8                  CP      '8'!80H           ;Menu request?
31FD CA9A32                JP      Z,MENU
                   ;
                           IF      @MOD4
3200 FEA8                  CP      '('!80H           ;Local clear screen?
                           ENDIF
                           IF      @MOD2
```

Program Code Section

```
                           CP      '*'+80H
                           ENDIF
3202 CA8F32                JP      Z,CLS
                   ;
                           IF      @MOD4
3205 FEA0                  CP      20H!80H          ;Clr-shf-0?
                           ENDIF
                           IF      @MOD2
                           CP      ')'+80H
                           ENDIF
3207 CA6535                JP      Z,DOSCMD         ;Do CMDR
                   ;
                           IF      @MOD4
320A FEBD                  CP      '='+80H          ;CK LDOS exit
                           ENDIF
                           IF      @MOD2
                           CP      '+'+80H
                           ENDIF
320C C27D32                JP      NZ,CMDERR
                   ;
                   ;
                   ;       Exit from LCOMM - Remove task vectors
                   ;
                   EXIT
                           IF      .NOT.BUFFRD
                           LD      C,8              ;Remove comm line scan task
320F                       @@RMTSK
                           LD      A,30
                           RST     40
                   ;
                           LD      C,9              ;Rmv printer task if used
320F                       @@RMTSK
                           LD      A,30
                           RST     40
                           ENDIF
                   ;
                           IF      BUFFRD
320F 11E437                LD      DE,CLDCB         ;Turn off wakeup feature
3212 FD210000              LD      IY,$-$
3214               OLDVEC  EQU     $-2              ;Restoring previous state
3216 0E04                  LD      C,4
3218                       @@CTL
3218 3E05                  LD      A,5
321A EF                    RST     40
                           ENDIF
                   ;
321B CDDF33                CALL    FR_OFF           ;Turn off any receive file
321E 112438                LD      DE,FR_FCB
3221 1A                    LD      A,(DE)
3222 CB7F                  BIT     7,A              ;Is it an open file?
3224 CA0030                JP      Z,$EXIT          ;Exit if not else
3227                       @@CLOSE                  ;  make sure it's closed
3227 3E3C                  LD      A,60
3229 EF                    RST     40
322A CA0030                JP      Z,$EXIT          ;Exit if no error
322D CD1A30                CALL    $ERROR
3230 C30A30                JP      $ABORT           ;Terminate
                   ;
                   ;       Query function ON or OFF
```

Program Code Section

```
                    ;
3233 CD5A32         QFUNC   CALL    QONOFF          ;Get On or Off response
3236 77                     LD      (HL),A          ;Save which one
3237 C9                     RET
                    ;
                    ;       Query *CL on or off
                    ;
3238 CD5A32         QCL     CALL    QONOFF
323B 77                     LD      (HL),A
323C B7                     OR      A               ;On or off?
323D C8                     RET     Z               ;Quit if off
323E 324534                 LD      (TASK8B+1),A    ;Force CL-send on as well
3241 C9                     RET
                    ;
                    ;       Query handshake on or off
                    ;
3242 D5             QSHAKE  PUSH    DE
3243                        @@KEY                   ;Get one key
3243 3EØ1                   LD      A,1
3245 EF                     RST     4Ø
3246 D1                     POP     DE
3247 A7                     AND     A               ;Be sure flags are set
3248 FA5132                 JP      M,QSHAKE1       ;Go if PF key
324B 326B34                 LD      (AUTXOFF+1),A   ;Save key as auto XOFF
324E 36FF                   LD      (HL),ØFFH       ;Turn on handshake
325Ø C9                     RET
3251 CD5F32         QSHAKE1 CALL    QONOFF1         ;Parse ON or OFF
3254 77                     LD      (HL),A          ;Turn on or off
3255 AF                     XOR     A               ;Turn off auto XOFF
3256 326B34                 LD      (AUTXOFF+1),A
3259 C9                     RET
325A D5             QONOFF  PUSH    DE              ;Hang on to register
325B                        @@KEY                   ;Get the operand key
325B 3EØ1                   LD      A,1
325D EF                     RST     4Ø
325E D1                     POP     DE              ;Restore the register
325F                QONOFF1 EQU     $
                            IF      @MOD4
325F FEAD                   CP      '-'+8ØH         ;Ck OFF
                            ENDIF
                            IF      @MOD2
                            CP      '='+8ØH
                            ENDIF
3261 2821                   JR      Z,TURNOF        ;  and go if off
                            IF      @MOD4
3263 FEBA                   CP      ':'+8ØH         ;Ck ON
                            ENDIF
                            IF      @MOD2
                            CP      '-'+8ØH
                            ENDIF
3265 281F                   JR      Z,TURNON        ;  and go if on
3267 E3             POPERR  EX      (SP),HL         ;Discard ret address
3268 E1                     POP     HL
3269 FEB9                   CP      '9'+8ØH         ;Ck ID
326B CA9633                 JP      Z,FILID
                    ;
326E FEBØ                   CP      'Ø'+8ØH         ;Ck RESET
327Ø CA4C33                 JP      Z,FILRES
                    ;
```

Program Code Section

```
3273 FEA5      08240          CP      '%'+80H          ;Ck REWIND
3275 CA8433    08250          JP      Z,FILREW
               08260 ;
               08270          IF      @MOD4
3278 FEA6      08280          CP      '&'+80H          ;Ck PEOF
               08290          ENDIF
               08300          IF      @MOD2
               08310          CP      '^'+80H
               08320          ENDIF
327A CA8D33    08330          JP      Z,FILEOF
               08340 ;
327D 218D37    08350 CMDERR   LD      HL,CMDERR$       ;None of above, dsply
3280           08360          @@DSPLY                  ;  "Unacceptable command...
               00039          IFEQ    00H,1
               00040          LD      HL,
               00041          ENDIF
3280 3E0A      00042          LD      A,10
3282 EF        00043          RST     40
3283 C9        08370          RET
               08380 ;
               08390 ;        Process OFF
               08400 ;
3284 AF        08410 TURNOF   XOR     A                ;Off = 0
3285 C9        08420          RET
               08430 ;
               08440 ;        Process ON
               08450 ;
3286 EB        08460 TURNON   EX      DE,HL            ;Shift "FCB" to HL
3287 CB7E      08470          BIT     7,(HL)           ;FCB on or non-file?
3289 EB        08480          EX      DE,HL            ;If non-file, HL now
328A 3EFF      08490          LD      A,0FFH           ;  points to X'0000'
328C C0        08500          RET     NZ               ;  which contains X'F3'
328D 18D8      08510          JR      POPERR           ;Is an error
               08520 ;
               08530 ;        Process Clear Screen
               08540 ;
328F 0E1C      08550 CLS      LD      C,1CH            ;Cursor home
3291           08560          @@DSP
3291 3E02      00044          LD      A,2
3293 EF        00045          RST     40
3294 0E1F      08570          LD      C,1FH            ;Clear to end-of-frame
3296           08580          @@DSP
3296 3E02      00046          LD      A,2
3298 EF        00047          RST     40
3299 C9        08590          RET
               08600 ;
               08610 ;        Process MENU
               08620 ;
329A           08630 MENU     EQU     $
329A 21E635    08640          LD      HL,STAT1         ;Clear top row status
329D 11E735    08650          LD      DE,STAT1+1       ;1st char always a space
32A0 014200    08660          LD      BC,66
32A3 EDB0      08670          LDIR
32A5 210537    08680          LD      HL,STAT2         ;Clear bottom row status
32A8 110637    08690          LD      DE,STAT2+1
32AB 0E26      08700          LD      C,38
32AD EDB0      08710          LDIR
32AF 060F      08720          LD      B,15             ;Init loop count
32B1 212D37    08730          LD      HL,STATAB        ;Words where status stored
```

Program Code Section

```
32B4 5E         Ø874Ø STATLP1 LD      E,(HL)          ;P/u lo-switch
32B5 23         Ø875Ø         INC     HL
32B6 56         Ø876Ø         LD      D,(HL)          ;P/u hi-switch
32B7 23         Ø877Ø         INC     HL
32B8 7E         Ø878Ø         LD      A,(HL)          ;P/u lo-stuff
32B9 23         Ø879Ø         INC     HL
32BA E5         Ø88ØØ         PUSH    HL              ;Save pointer
32BB 66         Ø881Ø         LD      H,(HL)          ;P/u hi-stuff
32BC 6F         Ø882Ø         LD      L,A             ;Xfer lo-stuff
32BD 1A         Ø883Ø         LD      A,(DE)          ;Get status
32BE B7         Ø884Ø         OR      A               ;Active or not?
32BF 28Ø2       Ø885Ø         JR      Z,$+4           ;Go if not
32C1 362A       Ø886Ø         LD      (HL),'*'        ;  else stuf an '*'
32C3 E1         Ø887Ø         POP     HL              ;Rcvr pointer
32C4 23         Ø888Ø         INC     HL              ;Bump to next pos
32C5 1ØED       Ø889Ø         DJNZ    STATLP1
32C7 1A         Ø89ØØ         LD      A,(DE)          ;P/u shake again
32C8 B7         Ø891Ø         OR      A
32C9 28ØD       Ø892Ø         JR      Z,STATLP2       ;Go if off
32CB 3A6B34     Ø893Ø         LD      A,(AUTXOFF+1)   ;Check if xoff char set
32CE B7         Ø894Ø         OR      A
32CF 28Ø7       Ø895Ø         JR      Z,STATLP2       ;Skip if not special char
32D1 212536     Ø896Ø         LD      HL,STAT1+63     ;Auto x-off char posn
32D4 4F         Ø897Ø         LD      C,A
32D5            Ø898Ø         @@HEX8                  ;Cvrt to ASCII for display
32D5 3E62       ØØØ48         LD      A,98
32D7 EF         ØØØ49         RST     4Ø
32D8 21E535     Ø899Ø STATLP2 LD      HL,MNUMSG       ;Ptr to Comm menu
32DB            Ø9ØØØ         @@DSPLY                 ;Display prelim status
                ØØØ5Ø         IFEQ    ØØH,1
                ØØØ51         LD      HL,
                ØØØ52         ENDIF
32DB 3EØA       ØØØ53         LD      A,1Ø
32DD EF         ØØØ54         RST     4Ø
32DE 21Ø438     Ø9Ø1Ø         LD      HL,FS_FCB       ;FS file open?
32E1 CB7E       Ø9Ø2Ø         BIT     7,(HL)
32E3 281C       Ø9Ø3Ø         JR      Z,STATLP3       ;Go if closed
32E5 114438     Ø9Ø4Ø         LD      DE,DUMMY        ;Recover its name w/o
32E8 D5         Ø9Ø5Ø         PUSH    DE              ;   changing the FCB
32E9 Ø12ØØØ     Ø9Ø6Ø         LD      BC,32
32EC EDBØ       Ø9Ø7Ø         LDIR                    ;  by creating a duplicate
32EE D1         Ø9Ø8Ø         POP     DE              ;  open FCB
32EF ED4B4A38   Ø9Ø9Ø         LD      BC,(DUMMY+6)    ;Get drive and DEC
32F3 D5         Ø91ØØ         PUSH    DE
32F4            Ø911Ø         @@FNAME                 ;Call for name recover
32F4 3E5Ø       ØØØ55         LD      A,8Ø
32F6 EF         ØØØ56         RST     4Ø
32F7 216937     Ø912Ø         LD      HL,FSNAME$      ;Output "FS-SPEC: "
32FA            Ø913Ø         @@DSPLY
                ØØØ57         IFEQ    ØØH,1
                ØØØ58         LD      HL,
                ØØØ59         ENDIF
32FA 3EØA       ØØØ6Ø         LD      A,1Ø
32FC EF         ØØØ61         RST     4Ø
32FD E1         Ø914Ø         POP     HL              ;Rcvr fcb pointer and
32FE            Ø915Ø         @@DSPLY                 ;  display the filespec
                ØØØ62         IFEQ    ØØH,1
                ØØØ63         LD      HL,
                ØØØ64         ENDIF
```

Program Code Section

```
32FE 3EØA      ØØØ65          LD      A,1Ø
33ØØ EF        ØØØ66          RST     4Ø
33Ø1 212438    Ø916Ø STATLP3  LD      HL,FR_FCB       ;Is the FR file open?
33Ø4 CB7E      Ø917Ø          BIT     7,(HL)
33Ø6 281C      Ø918Ø          JR      Z,STATLP4       ;Go if closed
33Ø8 114438    Ø919Ø          LD      DE,DUMMY        ;Similar to above
33ØB D5        Ø92ØØ          PUSH    DE
33ØC Ø12ØØØ    Ø921Ø          LD      BC,32
33ØF EDBØ      Ø922Ø          LDIR                    ;Create a duplicate FCB
3311 D1        Ø923Ø          POP     DE
3312 ED4B4A38  Ø924Ø          LD      BC,(DUMMY+6)    ;P/u Drive & DEC
3316 D5        Ø925Ø          PUSH    DE
3317           Ø926Ø          @@FNAME                 ;Call for name recover
3317 3E5Ø      ØØØ67          LD      A,8Ø
3319 EF        ØØØ68          RST     4Ø
331A 217337    Ø927Ø          LD      HL,FRNAME$      ;"FR-SPEC:"
331D           Ø928Ø          @@DSPLY
               ØØØ69          IFEQ    ØØH,1
               ØØØ7Ø          LD      HL,
               ØØØ71          ENDIF
331D 3EØA      ØØØ72          LD      A,1Ø
331F EF        ØØØ73          RST     4Ø
332Ø E1        Ø929Ø          POP     HL              ;P/u name start
3321           Ø93ØØ          @@DSPLY                 ;  and dsply it
               ØØØ74          IFEQ    ØØH,1
               ØØØ75          LD      HL,
               ØØØ76          ENDIF
3321 3EØA      ØØØ77          LD      A,1Ø
3323 EF        ØØØ78          RST     4Ø
3324 3AAD38    Ø931Ø STATLP4  LD      A,(FREEPG)      ;How much buffer left
3327 ØF        Ø932Ø          RRCA                    ;Divide by 4 to show
3328 ØF        Ø933Ø          RRCA                    ;  in K
3329 E63F      Ø934Ø          AND     3FH             ;No bit 7 nor 6
332B 218937    Ø935Ø          LD      HL,PAGSPR$+1Ø   ;Where to stuff
332E Ø6FF      Ø936Ø          LD      B,-1            ;Init to count 1Ø's
333Ø Ø4        Ø937Ø CVD1     INC     B
3331 D6ØA      Ø938Ø          SUB     1Ø              ;How many tens?
3333 3ØFB      Ø939Ø          JR      NC,CVD1         ;Go if no more
3335 F5        Ø94ØØ CVD2     PUSH    AF              ;Save remainder
3336 78        Ø941Ø          LD      A,B             ;P/u tens
3337 B7        Ø942Ø          OR      A               ;Was it zero tens?
3338 Ø62Ø      Ø943Ø          LD      B,' '           ;Init for space
333A 28Ø2      Ø944Ø          JR      Z,$+4           ;Go if no tens
333C Ø63Ø      Ø945Ø          LD      B,'Ø'           ;Init for ASCII
333E 8Ø        Ø946Ø          ADD     A,B             ;Convert to ASCII
333F 77        Ø947Ø          LD      (HL),A          ;Stuff & bump
334Ø 23        Ø948Ø          INC     HL
3341 F1        Ø949Ø          POP     AF              ;Get remainder
3342 C63A      Ø95ØØ          ADD     A,3AH           ;Adjust units place
3344 77        Ø951Ø          LD      (HL),A
3345 217F37    Ø952Ø          LD      HL,PAGSPR$      ;"Memory:   K"
3348           Ø953Ø          @@DSPLY
               ØØØ79          IFEQ    ØØH,1
               ØØØ8Ø          LD      HL,
               ØØØ81          ENDIF
3348 3EØA      ØØØ82          LD      A,1Ø
334A EF        ØØØ83          RST     4Ø
334B C9        Ø954Ø          RET
               Ø955Ø ;
```

Program Code Section

```
                 ;       Process RESET of a "device"
                 ;
334C 78          FILRES  LD      A,B             ;Check if a device vector
334D B1                  OR      C               ;  was passed
334E CA7D32              JP      Z,CMDERR        ;Go if not - is error
3351 7A                  LD      A,D             ;Check for a possible
3352 B3                  OR      E               ;  FCB for disk
3353 2016                JR      NZ,FILR4        ;Go if disk else device
                 ;
                 ;       Reset the page buffer(s) for the device
                 ;
3355 F3          FILR1   DI                      ;No interrupts until done
3356 60                  LD      H,B             ;Xfer vector table entry
3357 69                  LD      L,C             ;  to grab put/get index
3358 4E                  LD      C,(HL)          ;P/u the PUT pointer
3359 23                  INC     HL              ;  and make the GET
335A 46                  LD      B,(HL)          ;  pointer equal so
335B 23                  INC     HL              ;  buffer contents show
335C 71                  LD      (HL),C          ;  as empty
335D 23                  INC     HL
335E 7E                  LD      A,(HL)          ;P/u the GET pointer to
335F 70                  LD      (HL),B          ;  check if in same page
3360 B8          FILR2   CP      B               ;Is put/get in same page?
3361 2806                JR      Z,FILR3         ;Go if it is
3363 67                  LD      H,A             ;  else set up to free this
3364 CD4E35              CALL    FNPIU           ;  page by finding next
3367 18F7                JR      FILR2           ;Loop until next = 1st
3369 FB          FILR3   EI                      ;Interrupts back on
336A C9                  RET
                 ;
                 ;       Reset a file device
                 ;
336B 212438      FILR4   LD      HL,FR_FCB       ;Turn off the FR or FS
336E AF                  XOR     A
336F ED52                SBC     HL,DE           ;Is this the FR?
3371 214130              LD      HL,FSSW+1
3374 2006                JR      NZ,OFFS
3376 324831              LD      (FRIOSW+1),A    ;Turn off FR IO to disk
3379 212E31              LD      HL,FRSW+1       ;Turn off FR to buffer
337C 77          OFFS    LD      (HL),A          ;Turn off FR or FS
337D                     @@CLOSE                 ;Close the file
337D 3E3C                LD      A,60
337F EF                  RST     40
3380 C41A30              CALL    NZ,$ERROR       ;Show any close error
3383 C9                  RET
                 ;
                 ;       Process REWIND
                 ;
3384 7A          FILREW  LD      A,D             ;Rewind the specified
3385 B3                  OR      E               ;  file (FCB given) if
3386 CA7D32              JP      Z,CMDERR        ;  it is in use
3389                     @@REW
3389 3E44                LD      A,68
338B EF                  RST     40
338C C9                  RET
                 ;
                 ;       Process PEOF
                 ;
338D 7A          FILEOF  LD      A,D             ;Check if a file device
```

Program Code Section

```
338E B3        10110          OR      E               ;  was specified
338F CA7D32    10120          JP      Z,CMDERR        ;Go if not - is error
3392           10130          @@PEOF                  ;  else position to end
3392 3E41      00088          LD      A,65
3394 EF        00089          RST     40
3395 C9        10140          RET
               10150 ;
               10160 ;        Process ID request
               10170 ;
3396 7A        10180 FILID    LD      A,D             ;Bad command if not
3397 B3        10190          OR      E               ;  FS or FR specified
3398 CA7D32    10200          JP      Z,CMDERR        ;Go on error
339B 1A        10210          LD      A,(DE)          ;Make sure that it is
339C 07        10220          RLCA                    ;  not already open
339D 3834      10230          JR      C,NOTNOW        ;CF=already open
339F D5        10240          PUSH    DE
33A0 DDE5      10250          PUSH    IX              ;Save buffer pointer
33A2 21C335    10260          LD      HL,FILEPMT      ;Prompt for filespec
33A5           10270          @@DSPLY
               00090          IFEQ    00H,1
               00091          LD      HL,
               00092          ENDIF
33A5 3E0A      00093          LD      A,10
33A7 EF        00094          RST     40
33A8 E1        10280          POP     HL              ;Take file name
33A9 01001F    10290          LD      BC,31<8         ;31 chars max
33AC           10300          @@KEYIN                 ;Get the filespec
33AC 3E09      00095          LD      A,9
33AE EF        00096          RST     40
33AF F5        10310          PUSH    AF              ;Save flag state
33B0 0E0E      10320          LD      C,0EH           ;Turn the cursor back on
33B2           10330          @@DSP
33B2 3E02      00097          LD      A,2
33B4 EF        00098          RST     40
33B5 F1        10340          POP     AF              ;Rcvr KEYIN exit state
33B6 D1        10350          POP     DE
33B7 D8        10360          RET     C               ;Ret if BREAK from KEYIN
33B8 E5        10370          PUSH    HL              ;Save ptr to buffer
33B9           10380          @@FSPEC                 ;Fetch & parse filespec
33B9 3E4E      00099          LD      A,78
33BB EF        00100          RST     40
33BC 210438    10390          LD      HL,FS_FCB       ;Ck if FILID req from
33BF AF        10400          XOR     A               ;  FS or FR
33C0 ED52      10410          SBC     HL,DE           ;What's the FCB?
33C2 E1        10420          POP     HL              ;Recover buffer
33C3 0600      10430          LD      B,0             ;LRL=256
33C5 2005      10440          JR      NZ,FILFR        ;Go if req from FR
33C7 CD0F30    10450          CALL    $OPEN           ;Only OPEN a FS
33CA 1803      10460          JR      $+5             ;Branch around INIT
33CC           10470 FILFR    @@INIT                  ;Open the receive file
33CC 3E3A      00101          LD      A,58
33CE EF        00102          RST     40
33CF C41A30    10480          CALL    NZ,$ERROR       ;Show any open error
33D2 C9        10490          RET
               10500 ;
33D3 21D135    10510 NOTNOW   LD      HL,OPENMSG      ;"File already open"
33D6           10520          @@DSPLY                 ;Show why ID failed
               00103          IFEQ    00H,1
               00104          LD      HL,
```

Program Code Section

```
                ØØ1Ø5         ENDIF
33D6 3EØA       ØØ1Ø6         LD      A,1Ø
33D8 EF         ØØ1Ø7         RST     4Ø
33D9 C9         1Ø53Ø         RET
                1Ø54Ø ;
                1Ø55Ø ;       Routines to turn off file devices
                1Ø56Ø ;
33DA AF         1Ø57Ø FS_OFF  XOR     A               ;File send
33DB 32413Ø     1Ø58Ø         LD      (FSSW+1),A
33DE C9         1Ø59Ø         RET
33DF AF         1Ø6ØØ FR_OFF  XOR     A               ;File receive
33EØ 322E31     1Ø61Ø         LD      (FRSW+1),A
33E3 C9         1Ø62Ø         RET
33E4 AF         1Ø63Ø FRIO_OFF        XOR     A       ;Dump to disk
33E5 324831     1Ø64Ø         LD      (FRIOSW+1),A
33E8 C9         1Ø65Ø         RET
                1Ø66Ø ;
                1Ø67Ø ;       Call various tasks (on each main loop)
                1Ø68Ø ;
33E9 F3         1Ø69Ø TASKS   DI
                1Ø7ØØ ;
                1Ø71Ø         IF      .NOT.BUFFRD     ;W fcn does this if bfrd
                1Ø72Ø         CALL    TASK8A          ;Try to receive from *CL
                1Ø73Ø         ENDIF
                1Ø74Ø ;
33EA CD4434     1Ø75Ø         CALL    TASK8B          ;Try to send to *CL
33ED FB         1Ø76Ø         EI
33EE CD7434     1Ø77Ø         CALL    TASKK           ;Allow interrupts here
                1Ø78Ø         IF      .NOT.BUFFRD
                1Ø79Ø         DI                      ;If RS232 does not interrupt
                1Ø8ØØ         ENDIF                   ;Printer must be task
33F1 CDB834     1Ø81Ø         CALL    TASK9
33F4 FB         1Ø82Ø         EI
33F5 C34731     1Ø83Ø         JP      FRIOSW          ;Check on dump to disk
                1Ø84Ø ;
                1Ø85Ø ;       INTERRUPT TASK 8
                1Ø86Ø ; WO/buffer    A is done once per main loop + int rate
                1Ø87Ø ;              B is done once per main loop + int rate
                1Ø88Ø ; W/buffer     A is done by wakeup feature + int rate
                1Ø89Ø ;              B is done once per main loop + int rate
                1Ø9ØØ ;
                1Ø91Ø         IF      .NOT.BUFFRD
                1Ø92Ø TCB8    DW      TASK8
                1Ø93Ø TASK8   CALL    TASK8A
                1Ø94Ø         JP      TASK8B
                1Ø95Ø         ENDIF
                1Ø96Ø ;
33F8 F3         1Ø97Ø TASK8A  DI
33F9 3EFF       1Ø98Ø         LD      A,ØFFH          ;CL recv On/Off
33FB B7         1Ø99Ø         OR      A
33FC C8         11ØØØ         RET     Z               ;Done if CL recv off
                11Ø1Ø ;
                11Ø2Ø ;       @GET handler to keep interrupts off if possible
                11Ø3Ø ;
33FD 11E437     11Ø4Ø         LD      DE,CLDCB        ;=> OPEN DCB
34ØØ 62         11Ø5Ø FNDDVR  LD      H,D             ;Xfer to HL
34Ø1 6B         11Ø6Ø         LD      L,E
34Ø2 7E         11Ø7Ø         LD      A,(HL)          ;Get DCB type
34Ø3 CB6F       11Ø8Ø         BIT     5,A             ;Is it linked?
```

```
3405 2013              JR      NZ,LNKD         ;Need CHNIO if so
3407 23                INC     HL              ;=>address field of DCB
3408 5E                LD      E,(HL)          ;If routed, address is
3409 23                INC     HL              ;  next DCB to use
340A 56                LD      D,(HL)          ;  else EP of driver
340B CB67              BIT     4,A             ;Z = not routed
340D 20F1              JR      NZ,FNDDVR       ;Loop till not routed
340F E608              AND     00001000B       ;Can't talk to NIL device
3411 C0                RET     NZ
3412 EB                EX      DE,HL           ;Address to HL
3413 111D34            LD      DE,RETADD       ;Put RET address on stack
3416 D5                PUSH    DE              ;
3417 FE02              CP      2               ;Set C,NZ for input request
3419 E9                JP      (HL)            ;Go to driver
                ;
341A            LNKD   @@GET                   ;Use SVC if LINKED
341A 3E03              LD      A,3
341C EF                RST     40
341D C0         RETADD RET     NZ              ;NZ means no char rcv'd
                ;
341E 0600       EIGHT  LD      B,0             ;Eight bit mode switch
3420 04                INC     B
3421 05                DEC     B
3422 2006              JR      NZ,XLTR1        ;Go if 8 bit
3424 E67F              AND     7FH             ;Strip bit 7
3426 C8                RET     Z               ;Always ignore nulls
3427 FE7F              CP      7FH             ;  & DELETE if not 8-bit
3429 C8                RET     Z
                ;
                ;      Do XLATER after stripping high bit
                ;
342A FE00       XLTR1  CP      $-$             ;Character to translate?
342C 2002              JR      NZ,TSTNUL       ;Go if not a match
342E 3E00       XLTR2  LD      A,$-$           ;Replace with xlated char
                ;
                ;      NULL Parm now only affects 8-bit mode
                ;
3430 B7         TSTNUL OR      A               ;Is char a null?
3431 2005              JR      NZ,KEEPCH       ;Go if not
3433 06FF       ACCNUL LD      B,0FFH          ;Default to accept nulls
3435 04                INC     B               ;NZ=nulls wanted
3436 05                DEC     B               ;Z=don't accept nulls
3437 C8                RET     Z
                ;
3438 DDE5       KEEPCH PUSH    IX              ;Place in CL input buf
343A DD219D38          LD      IX,CLREC
343E CDD634            CALL    OUTPUT          ;Out to the buffer if
3441 DDE1              POP     IX              ;  non-null or want nulls
3443 C9                RET
                ;
3444 3EFF       TASK8B LD      A,0FFH          ;CL send On/Off for
3446 B7                OR      A               ;  handshaking
3447 C8                RET     Z
3448 0E00              LD      C,0             ;Now xmit a CTL0 to
344A 11E437            LD      DE,CLDCB        ;Ck the status of the
344D                   @@CTL                   ;  CL
344D 3E05              LD      A,5
344F EF                RST     40
3450 C0                RET     NZ              ;Indicates not ready
```

Program Code Section

```
3451 ØEØØ     1164Ø FRCPUT  LD      C,$-$           ;Force a char out?
3453 AF       1165Ø         XOR     A               ;Clear it after p/u
3454 325234   1166Ø         LD      (FRCPUT+1),A
3457 B1       1167Ø         OR      C               ;Check original status
3458 2ØØD     1168Ø         JR      NZ,FRCIT        ;Go if force on
345A DDE5     1169Ø         PUSH    IX
345C DD21A138 117ØØ         LD      IX,CLSEND       ;Do we have a char to
346Ø CD1635   1171Ø         CALL    BUFGET          ;  send to the CL?
3463 DDE1     1172Ø         POP     IX
3465 C8       1173Ø         RET     Z               ;RET if not
3466 4F       1174Ø         LD      C,A             ;Save character
3467          1175Ø FRCIT   @@PUT                   ;Put it out
3467 3EØ4     ØØ112         LD      A,4
3469 EF       ØØ113         RST     4Ø
              1176Ø ;
346A 3EØØ     1177Ø AUTXOFF LD      A,Ø             ;Check for auto XOFF
346C B7       1178Ø         OR      A               ;On?
346D C8       1179Ø         RET     Z               ;Quit if not
346E 91       118ØØ         SUB     C               ;Matched char?
346F CØ       1181Ø         RET     NZ              ;Quit if not
347Ø 324534   1182Ø         LD      (TASK8B+1),A    ;Pause xmit (XOFF)
3473 C9       1183Ø         RET
              1184Ø ;
3474          1185Ø TASKK   @@KBD                   ;Scan the keyboard
3474 3EØ8     ØØ114         LD      A,8
3476 EF       ØØ115         RST     4Ø
3477 CØ       1186Ø         RET     NZ              ;Error (or no key depressed)
3478 FE8Ø     1187Ø         CP      BREAK           ;Ck for brk 1st
347A 2815     1188Ø         JR      Z,ISBRK         ;Go on a Break
347C B7       1189Ø         OR      A               ;Then for high bit set
347D FA6831   119ØØ         JP      M,CMDKEY        ;Go if FN key
348Ø Ø6FF     1191Ø KISW    LD      B,ØFFH          ;KI On/Off
3482 Ø4       1192Ø         INC     B
3483 Ø5       1193Ø         DEC     B
3484 C8       1194Ø         RET     Z               ;Ret if KI is off
3485 DDE5     1195Ø NOTBRK  PUSH    IX
3487 DD219538 1196Ø         LD      IX,KIVCTR       ;  else put key into
348B CDAE38   1197Ø         CALL    OUTPGM          ;  the output buffer
348E DDE1     1198Ø         POP     IX
349Ø C9       1199Ø         RET
              12ØØØ ;
              12Ø1Ø ISBRK
3491 F3       12Ø2Ø         DI
3492 11E437   12Ø3Ø         LD      DE,CLDCB        ;Pt to *CL
3495 ØEØ1     12Ø4Ø         LD      C,1             ;Send CTL 1, a
3497          12Ø5Ø         @@CTL                   ;  Break request
3497 3EØ5     ØØ116         LD      A,5
3499 EF       ØØ117         RST     4Ø
349A FB       12Ø6Ø         EI
349B Ø1A138   12Ø7Ø         LD      BC,CLSEND
349E CD5533   12Ø8Ø         CALL    FILR1           ;Reset the CL buffer
34A1 CDDA33   12Ø9Ø         CALL    FS_OFF          ;Turn off the *FS
34A4 Ø1ØØ2Ø   121ØØ         LD      BC,2ØØØH        ;Time delay
34A7          1211Ø         @@PAUSE
34A7 3E1Ø     ØØ118         LD      A,16
34A9 EF       ØØ119         RST     4Ø
34AA          1212Ø         @@CKBRKC                ;Reset the break bit
34AA 3E6A     ØØ12Ø         LD      A,1Ø6
34AC EF       ØØ121         RST     4Ø
```

Program Code Section

```
34AD 0E00                LD      C,0             ;Init the character
34AF 11E437              LD      DE,CLDCB        ;P/u the CL DCB
34B2 F3                  DI
34B3                     @@PUT                   ;Send the 0
34B3 3E04                LD      A,4
34B5 EF                  RST     40
34B6 FB                  EI
34B7 C9                  RET
              ;
              ;          INTERRUPT TASK 9
              ;          Only if RS232 does not interrupt
              ;
                         IF      .NOT.BUFFRD
              TCB9       DW      TASK9           ;Task ept
                         ENDIF
34B8 0603     TASK9      LD      B,3             ;Max chars/pass
34BA 0E00     PRLOOP     LD      C,0             ;Test printer status
34BC 110000              LD      DE,$-$          ;PDCB$
34BD          PRDCB      EQU     $-2
34BF                     @@CTL                   ;Check the status
34BF 3E05                LD      A,5
34C1 EF                  RST     40
34C2 C0                  RET     NZ              ;Ret if unavailable
34C3 DDE5                PUSH    IX
34C5 DD219938            LD      IX,PRVCTR       ;Get char from printer
                         IF      BUFFRD
34C9 CDB438              CALL    PGMGET
                         ELSE
                         CALL    BUFGET          ;Buffer if available
                         ENDIF
34CC DDE1                POP     IX
34CE C8                  RET     Z               ;None to get, back
34CF 4F                  LD      C,A
34D0                     @@PRT                   ;Output to printer
34D0 3E06                LD      A,6
34D2 EF                  RST     40
34D3 10E5                DJNZ    PRLOOP          ;Loop if more
34D5 C9                  RET
              ;
              ;          Common routine to stuff various buffers
              ;
34D6 DD6E00   OUTPUT     LD      L,(IX)          ;P/u pointer to
34D9 DD6601              LD      H,(IX+1)        ;  buffer PUT
34DC 77                  LD      (HL),A          ;Write char into buffer
34DD DD3400              INC     (IX)            ;Bump buffer pointer
34E0 C0                  RET     NZ              ;Go if still in same page
34E1 CD3635              CALL    NEXTAP          ;Find next avail page
34E4 2810                JR      Z,DUMPCHR       ;Go if no pages available
34E6 DD7701              LD      (IX+1),A        ;Set index to new page
34E9 21AD38              LD      HL,FREEPG       ;Reduce the amount of
34EC 35                  DEC     (HL)            ;  free pages
34ED 3E07                LD      A,7             ;Less than 2K available?
34EF BE                  CP      (HL)
34F0 D8                  RET     C               ;  & return with NZ
34F1 322430              LD      (MAINLP+1),A    ;Set flag for warning
34F4 B7                  OR      A               ;Ensure NZ return
34F5 C9                  RET
              ;
              ;          No more pages available - keep last page
```

Program Code Section

```
              ;
34F6 DD35ØØ   DUMPCHR DEC     (IX)            ;Dump character and
34F9 AF               XOR     A               ;  return
34FA C9               RET
              ;
              ;       The following code is not executed, as it is too
              ;        slow at rates >= 12ØØ baud because interuppts are on.
              ;       DE must be loaded with KIVCTR.
              ;
34FB ØØ               DB      Ø
34FC DDE5             PUSH    IX              ;Dev requesting the output
34FE E1               POP     HL
34FF AF               XOR     A               ;The difference will be
35ØØ ED52             SBC     HL,DE           ;  the offset into the
35Ø2 116A3E           LD      DE,DEVICE$      ;  name table
35Ø5 19               ADD     HL,DE
35Ø6 Ø1Ø4ØØ           LD      BC,4
35Ø9 11893E           LD      DE,OVRRUN$+3
35ØC EDBØ             LDIR
35ØE 21863E           LD      HL,OVRRUN$      ;Display the buffer
3511                  @@DSPLY                 ;  overrun error
                      IFEQ    ØØH,1
                      LD      HL,
                      ENDIF
3511 3EØA             LD      A,1Ø
3513 EF               RST     4Ø
3514 AF               XOR     A               ;  reuse current page
3515 C9               RET
              ;
              ;       Check for character available in dynamic buffer
              ;
3516 DD6EØ2   BUFGET  LD      L,(IX+2)        ;P/u pointer to next
3519 DD66Ø3           LD      H,(IX+3)        ;  buffer GET
351C 7D               LD      A,L             ;Check on in=out lo-order
351D DDBEØØ           CP      (IX)
352Ø 2ØØ5             JR      NZ,INNEOUT      ;Go if in not equal out
3522 7C               LD      A,H             ;Check on in=out hi-order
3523 DDBEØ1           CP      (IX+1)
3526 C8               RET     Z               ;Ret if none to i/o
              ;
              ;       Buffer is not empty - Get next character
              ;
3527 7E       INNEOUT LD      A,(HL)          ;Get a char from buffer
3528 DD34Ø2           INC     (IX+2)          ;Advance lo-order pointer
352B CØ               RET     NZ              ;Ret if still same page
352C F5               PUSH    AF              ;Save the character
352D CD4E35           CALL    FNPIU           ;Find next page in use
353Ø DD77Ø3           LD      (IX+3),A        ;Stuff new page index
3533 F1               POP     AF              ;Recover the character
3534 25               DEC     H               ;Set NZ return for rcvd
3535 C9               RET
              ;
              ;       Routine to find next available page buffer
              ;
3536 6C       NEXTAP  LD      L,H             ;Point to page buffer
3537 2639             LD      H,LINKS<-8      ;  index
3539 3AØØ39           LD      A,(LINKS)       ;Get next empty link
353C E5               PUSH    HL              ;Save this index pointer
353D 6F               LD      L,A             ;Point to new link
```

Program Code Section

```
353E 7E         132ØØ          LD      A,(HL)          ;Get what it links to
353F B7         1321Ø          OR      A               ;Test if none left
354Ø 2ØØ2       1322Ø          JR      NZ,GOTNAP       ;Go if still more
3542 E1         1323Ø          POP     HL              ;Restore reg & return
3543 C9         1324Ø          RET                     ;  with Z-flag for error
                1325Ø ;
                1326Ø ;        Found the next available page - set the links
                1327Ø ;
3544 32ØØ39     1328Ø GOTNAP   LD      (LINKS),A       ;Update new next avail
3547 7D         1329Ø          LD      A,L             ;Xfer index of new page
3548 E1         133ØØ          POP     HL              ;Rcvr pointer to index
3549 77         1331Ø          LD      (HL),A          ;  of old & link to new
354A 6F         1332Ø          LD      L,A             ;Repoint to new page's
354B 36ØØ       1333Ø          LD      (HL),Ø          ;  index & show it is
354D C9         1334Ø          RET                     ;  the last one
                1335Ø ;
                1336Ø ;        Find next page in use
                1337Ø ;
354E 3AAD38     1338Ø FNPIU    LD      A,(FREEPG)      ;Show one additional
3551 3C         1339Ø          INC     A               ;  page is free
3552 32AD38     134ØØ          LD      (FREEPG),A
3555 3AØ139     1341Ø          LD      A,(HIPAGE)      ;P/u highest page avail
3558 6F         1342Ø          LD      L,A             ;Set HL to its index
3559 7C         1343Ø          LD      A,H
355A 2639       1344Ø          LD      H,LINKS<-8      ;Show that page links to
355C 77         1345Ø          LD      (HL),A          ; the one we just emptied
355D 32Ø139     1346Ø          LD      (HIPAGE),A      ;Now update the new end
356Ø 6F         1347Ø          LD      L,A             ;Set HL to the emptied
3561 7E         1348Ø          LD      A,(HL)          ;  page, p/u what it
3562 36ØØ       1349Ø          LD      (HL),Ø          ;  linked to, & show old
3564 C9         135ØØ          RET                     ;  is end. Ret A=link
                1351Ø ;
                1352Ø ;        Execute a DOS command
                1353Ø ;
3565 21FF2F     1354Ø DOSCMD   LD      HL,BASE-1
3568 Ø6Ø1       1355Ø          LD      B,1             ;Set LOW$
356A            1356Ø          @@HIGH$
356A 3E64       ØØ133          LD      A,1ØØ
356C EF         ØØ134          RST     4Ø
356D 219A35     1357Ø          LD      HL,CMDPMT       ;Issue prompt
357Ø            1358Ø          @@DSPLY
                ØØ135          IFEQ    ØØH,1
                ØØ136          LD      HL,
                ØØ137          ENDIF
357Ø 3EØA       ØØ138          LD      A,1Ø
3572 EF         ØØ139          RST     4Ø
3573 Ø1ØØ5Ø     1359Ø          LD      BC,8Ø<8         ;Max characters
3576 214438     136ØØ          LD      HL,DUMMY        ;=>input buffer
3579            1361Ø          @@KEYIN                 ;Get command request
3579 3EØ9       ØØ14Ø          LD      A,9
357B EF         ØØ141          RST     4Ø
357C D8         1362Ø          RET     C               ;Back on Break
357D Ø4         1363Ø          INC     B
357E Ø5         1364Ø          DEC     B
357F C8         1365Ø          RET     Z               ;  or CR only
358Ø EB         1366Ø          EX      DE,HL
3581 21ØØØØ     1367Ø          LD      HL,$-$          ;Pt to CFLAG$
3582            1368Ø CFLAG    EQU     $-2
3584 CB46       1369Ø          BIT     Ø,(HL)          ;Get current status
```

Program Code Section

```
3586 E5         137ØØ          PUSH    HL
3587 F5         1371Ø          PUSH    AF               ;Save memory freeze status
3588 CBC6       1372Ø          SET     Ø,(HL)           ;Freeze memory
358A EB         1373Ø          EX      DE,HL
358B            1374Ø          @@CMNDR                  ;Do the command
358B 3E19       ØØ142          LD      A,25
358D EF         ØØ143          RST     4Ø
358E 21AB35     1375Ø          LD      HL,CMPLTD        ;Show cmd finished
3591            1376Ø          @@DSPLY
                ØØ144          IFEQ    ØØH,1
                ØØ145          LD      HL,
                ØØ146          ENDIF
3591 3EØA       ØØ147          LD      A,1Ø
3593 EF         ØØ148          RST     4Ø
3594 F1         1377Ø          POP     AF               ;Get the previous status
3595 E1         1378Ø          POP     HL               ;  and CFLAG$ location
3596 CØ         1379Ø          RET     NZ               ;Back if was set before
3597 CB86       138ØØ          RES     Ø,(HL)           ;  else restore it
3599 C9         1381Ø          RET
359A ØA         1382Ø CMDPMT   DB      LF,LF,'Enter command:',CR
     ØA 45 6E 74 65 72 2Ø 63
     6F 6D 6D 61 6E 64 3A ØD
35AB ØA         1383Ø CMPLTD   DB      LF,'Command completed',CR
     43 6F 6D 6D 61 6E 64 2Ø
     63 6F 6D 7Ø 6C 65 74 65
     64 ØD
                1384Ø ;
                1385Ø ;        Messages
                1386Ø ;
35BE 7B         1387Ø BRAKET   DB      '{  }',3         ;Brackets around hex byte
     2Ø 2Ø 7D Ø3
35C3 1D         1388Ø FILEPMT  DB      29,1Ø,'File name: ',3
     ØA 46 69 6C 65 2Ø 6E 61
     6D 65 3A 2Ø Ø3
35D1 1D         1389Ø OPENMSG  DB      29,1Ø,'File already open',CR
     ØA 46 69 6C 65 2Ø 61 6C
     72 65 61 64 79 2Ø 6F 7Ø
     65 6E ØD
35E5 ØA         139ØØ MNUMSG   DB      LF
35E6 2Ø         1391Ø STAT1    DB      '                                      '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø
36ØA 2Ø         1392Ø          DB      '                                      ',LF
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø ØA
362E 44         1393Ø          DB      'DUPLX ECHO  ECOLF ACCLF REWND PEOF  '
     55 5Ø 4C 58 2Ø 45 43 48
     4F 2Ø 2Ø 45 43 4F 4C 46
     2Ø 41 43 43 4C 46 2Ø 52
     45 57 4E 44 2Ø 5Ø 45 4F
     46 2Ø 2Ø
3652 2Ø         1394Ø          DB      ' DCC   CLS   8-B   CMD  HNDSH  EXIT',LF
     44 43 43 2Ø 2Ø 2Ø 43 4C
```

Program Code Section

```
     53 2Ø 2Ø 2Ø 38 2D 42 2Ø
     2Ø 2Ø 43 4D 44 2Ø 2Ø 48
     4E 44 53 48 2Ø 2Ø 45 58
     49 54 ØA
3676 3D        1395Ø          DB      '==1== ==2== ==3== ==4== ==5== ==6== '
     3D 31 3D 3D 2Ø 3D 3D 32
     3D 3D 2Ø 3D 3D 33 3D 3D
     2Ø 3D 3D 34 3D 3D 2Ø 3D
     3D 35 3D 3D 2Ø 3D 3D 36
     3D 3D 2Ø
369A 3D        1396Ø          DB      '==7== ==8== ==9== ==Ø== '
     3D 37 3D 3D 2Ø 3D 3D 38
     3D 3D 2Ø 3D 3D 39 3D 3D
     2Ø 3D 3D 3Ø 3D 3D 2Ø
               1397Ø          IF      @MOD4
36B2 3D        1398Ø          DB      '==:== ==-=='
     3D 3A 3D 3D 2Ø 3D 3D 2D
     3D 3D
               1399Ø          ENDIF
               14ØØØ          IF      @MOD2
               14Ø1Ø          DB      '==-== ====='
               14Ø2Ø          ENDIF
36BD ØA        14Ø3Ø          DB      LF
36BE 2Ø        14Ø4Ø          DB      ' *KI   *DO   *PR   *CL   *FS   *FR  '
     2A 4B 49 2Ø 2Ø 2Ø 2A 44
     4F 2Ø 2Ø 2Ø 2A 5Ø 52 2Ø
     2Ø 2Ø 2A 43 4C 2Ø 2Ø 2Ø
     2A 46 53 2Ø 2Ø 2Ø 2A 46
     52 2Ø 2Ø
36E2 2Ø        14Ø5Ø          DB      ' DTD   ???   ID    RES   ON    OFF',LF
     44 54 44 2Ø 2Ø 2Ø 3F 3F
     3F 2Ø 2Ø 2Ø 49 44 2Ø 2Ø
     2Ø 2Ø 52 45 53 2Ø 2Ø 2Ø
     4F 4E 2Ø 2Ø 2Ø 2Ø 4F 46
     46 ØA
37Ø5 2Ø        14Ø6Ø STAT2    DB      '                                     '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø
372A 2Ø        14Ø7Ø          DB      '  ',CR
     2Ø ØD
372D 8134      14Ø8Ø STATAB   DW      KISW+1,STAT2+2,DEVOUT+1,STAT2+8
     Ø737 1731 ØD37
3735 3B31      14Ø9Ø          DW      PUTPR+1,STAT2+14,TASK8B+1,STAT2+19
     1337 4534 1837
373D FA33      141ØØ          DW      TASK8A+2,STAT2+21,FSSW+1,STAT2+26
     1A37 413Ø 1F37
3745 2E31      1411Ø          DW      FRSW+1,STAT2+32,FRIOSW+1,STAT2+38
     2537 4831 2B37
374D 6A3Ø      1412Ø          DW      DPLXSW+1,STAT1+2,ECHOSW+1,STAT1+8
     E835 C93Ø EE35
3755 ØA31      1413Ø          DW      ECOLF,STAT1+14,ACCLFSW+1,STAT1+2Ø
     F435 DB3Ø FA35
375D 893Ø      1414Ø          DW      DSPCTRL+1,STAT1+38,EIGHT+1,STAT1+5Ø
     ØC36 1F34 1836
3765 A33Ø      1415Ø          DW      SHAKE+1,STAT1+61
     2336
```

Program Code Section

```
3769 46         1416Ø FSNAME$ DB      'FS-Spec: ',3
     53 2D 53 7Ø 65 63 3A 2Ø
     Ø3
3773 2Ø         1417Ø FRNAME$ DB      '  FR-Spec: ',3
     2Ø 46 52 2D 53 7Ø 65 63
     3A 2Ø Ø3
377F 2Ø         1418Ø PAGSPR$ DB      '  Memory:   K',CR
     2Ø 4D 65 6D 6F 72 79 3A
     2Ø 2Ø 2Ø 4B ØD
378D 2A         1419Ø CMDERR$ DB      '** Invalid command sequence **',CR
     2A 2Ø 49 6E 76 61 6C 69
     64 2Ø 63 6F 6D 6D 61 6E
     64 2Ø 73 65 71 75 65 6E
     63 65 2Ø 2A 2A ØD
37AC 57         1420Ø LILPG$  DB      'Warning! Less than 2K of buffer left '
     61 72 6E 69 6E 67 21 2Ø
     4C 65 73 73 2Ø 74 68 61
     6E 2Ø 32 4B 2Ø 6F 66 2Ø
     62 75 66 66 65 72 2Ø 6C
     65 66 74 2Ø
37D1 2Ø         1421Ø         DB      ' X-OFF transmitted',CR
     58 2D 4F 46 46 2Ø 74 72
     61 6E 73 6D 69 74 74 65
     64 ØD
                1422Ø ;
                1423Ø ;       File control blocks
                1424Ø ;
ØØ2Ø            1425Ø CLDCB   DS      32
ØØ2Ø            1426Ø FS_FCB  DS      32
ØØ2Ø            1427Ø FR_FCB  DS      32
ØØ51            1428Ø DUMMY   DS      81                 ;Used for dos cmd buffer also
                1429Ø ;
                1430Ø ;       Put/Get index pointers
                1431Ø ;
3895 ØØØØ       1432Ø KIVCTR  DW      Ø,Ø
     ØØØØ
3899 ØØØØ       1433Ø PRVCTR  DW      Ø,Ø
     ØØØØ
389D ØØØØ       1434Ø CLREC   DW      Ø,Ø
     ØØØØ
38A1 ØØØØ       1435Ø CLSEND  DW      Ø,Ø
     ØØØØ
38A5 ØØØØ       1436Ø FSVCTR  DW      Ø,Ø
     ØØØØ
38A9 ØØØØ       1437Ø FRVCTR  DW      Ø,Ø
     ØØØØ
ØØØ1            1438Ø FREEPG  DS      1
                1439Ø ;
                1440Ø ;       Routines  to buffer I/O in pgm loop
                1441Ø ;
38AE F3         1442Ø OUTPGM  DI
38AF CDD634     1443Ø         CALL    OUTPUT
38B2 FB         1444Ø         EI
38B3 C9         1445Ø         RET
38B4 F3         1446Ø PGMGET  DI
38B5 CD1635     1447Ø         CALL    BUFGET
38B8 FB         1448Ø         EI
38B9 C9         1449Ø         RET
                1450Ø ;
```

Program Code Section

```
             ;        Page buffer Link table
             ;
3900                  ORG     $<-8+1<+8
0001           LINKS  DS      1                ;Link to next available
0001           HIPAGE DS      1                ;Link to last available
0001                  DS      1                ;Init to 1st avail
0001                  DS      1                ;Init to last avail
00FC                  DS      252              ;Space for linkage tables
             ;
             ;        Transmit and Receive File buffers
             ;
0100           XMTBUF DS      256
0100           RCVBUF DS      256
             ;
3C00                  SUBTTL  '<COMM initialization code>'
```

COMM initialization code

```
3CØØ                *GET    LCOMMA:3                ;Initialization code
                    ;LCOMMA/ASM - COMM Initialization Code
                    ;
                    ;       Entry point to LCOMM
                    ;
                    LCOMM
3CØØ                        @@CKBRKC                ;Check for break
3CØØ 3E6A                   LD      A,1Ø6
3CØ2 EF                     RST     4Ø
3CØ3 28Ø4                   JR      Z,LCOMMA        ;Continue if not
3CØ5 21FFFF                 LD      HL,-1           ;  else ABORT
3CØ8 C9                     RET
                    ;
3CØ9 F3             LCOMMA  DI
3CØA ED73Ø43Ø               LD      (STACK),SP      ;Save for exit
3CØE E5                     PUSH    HL              ;Save ptr to CMD buffer
3CØF 21ØØØØ                 LD      HL,Ø
3C12                        @@BREAK                 ;Disable break vectoring
                            IFEQ    ØØH,1
                            LD      HL,
                            ENDIF
3C12 3E67                   LD      A,1Ø3
3C14 EF                     RST     4Ø
3C15 FB                     EI
3C16 21533D                 LD      HL,HELLO$       ;Issue the copyright
3C19                        @@DSPLY
                            IFEQ    ØØH,1
                            LD      HL,
                            ENDIF
3C19 3EØA                   LD      A,1Ø
3C1B EF                     RST     4Ø
3C1C E1                     POP     HL
3C1D 11E437                 LD      DE,CLDCB        ;Point to FCB
3C2Ø                        @@FSPEC                 ;Get the *CL spec
3C2Ø 3E4E                   LD      A,78
3C22 EF                     RST     4Ø
3C23 C24A3D                 JP      NZ,BADCL        ;Go error if none
3C26 1A                     LD      A,(DE)
3C27 FE2A                   CP      '*'             ;Ck for device spec
3C29 C24A3D                 JP      NZ,BADCL        ;Go if not a device
3C2C 11A23E                 LD      DE,PRMTBL$
3C2F                        @@PARAM                 ;Parse the parms
3C2F 3E11                   LD      A,17
3C31 EF                     RST     4Ø
3C32 F5                     PUSH    AF              ;Save status
3C33 C41A3Ø                 CALL    NZ,$ERROR       ;Display any error
3C36 F1                     POP     AF
3C37 C2ØA3Ø                 JP      NZ,$ABORT       ;  and then quit
                    ;
3C3A Ø6ØØ                   LD      B,Ø
3C3C 11E437                 LD      DE,CLDCB        ;Open the comm line
3C3F                        @@OPEN
3C3F 3E3B                   LD      A,59
3C41 EF                     RST     4Ø
3C42 F5                     PUSH    AF
3C43 C41A3Ø                 CALL    NZ,$ERROR       ;Show any open error
3C46 F1                     POP     AF
3C47 C2ØA3Ø                 JP      NZ,$ABORT       ;  and then quit
3C4A ØEØ2                   LD      C,2             ;INIT function for hardware
3C4C                        @@CTL                   ;Just in case
```

COMM initialization code

```
3C4C 3EØ5      ØØ167          LD      A,5
3C4E EF        ØØ168          RST     4Ø
3C4F 21D63D    15Ø9Ø          LD      HL,GETMNU$       ;How the user gets menu
3C52           151ØØ          @@DSPLY
               ØØ169          IFEQ    ØØH,1
               ØØ17Ø          LD      HL,
               ØØ171          ENDIF
3C52 3EØA      ØØ172          LD      A,1Ø
3C54 EF        ØØ173          RST     4Ø
3C55 AF        1511Ø          XOR     A
3C56 32Ø438    1512Ø          LD      (FS_FCB),A       ;Init FCB's to OFF
3C59 322438    1513Ø          LD      (FR_FCB),A
3C5C ED5BAØ3E  1514Ø          LD      DE,(PRNAME)      ;Load 'PR' backwards
3C6Ø           1515Ø          @@GTDCB
3C6Ø 3E52      ØØ174          LD      A,82
3C62 EF        ØØ175          RST     4Ø
3C63 22BD34    1516Ø          LD      (PRDCB),HL       ;Store address for @CTL
3C66           1517Ø          @@FLAGS                  ;Set up IY
3C66 3E65      ØØ176          LD      A,1Ø1
3C68 EF        ØØ177          RST     4Ø
3C69 FDE5      1518Ø          PUSH    IY
3C6B D1        1519Ø          POP     DE
3C6C 2112ØØ    152ØØ          LD      HL,'S'-'A'       ;Offset to SFLAG$
3C6F 19        1521Ø          ADD     HL,DE
3C7Ø 22113Ø    1522Ø          LD      (SFLG),HL        ;Store for later
3C73 21ØAØØ    1523Ø          LD      HL,'K'-'A'       ;Offset to KFLAG$
3C76 19        1524Ø          ADD     HL,DE
3C77 CB86      1525Ø          RES     Ø,(HL)           ;Be sure BREAK bit is off
3C79 21Ø2ØØ    1526Ø          LD      HL,'C'-'A'       ;CFLAG$
3C7C 19        1527Ø          ADD     HL,DE
3C7D 228235    1528Ø          LD      (CFLAG),HL
3C8Ø CB4E      1529Ø          BIT     1,(HL)           ;Doing CMNDR?
3C82 21ØØØØ    153ØØ          LD      HL,Ø
3C85 45        1531Ø          LD      B,L
3C86 28Ø1      1532Ø          JR      Z,$+3            ;Use LOW$ if CMNDR
3C88 Ø4        1533Ø          INC     B
3C89           1534Ø          @@HIGH$
3C89 3E64      ØØ178          LD      A,1ØØ
3C8B EF        ØØ179          RST     4Ø
3C8C 23        1535Ø          INC     HL               ;Available for use
3C8D 25        1536Ø          DEC     H                ;  by page buffers
3C8E 44        1537Ø          LD      B,H              ;Set B to highest usable
3C8F 21ØØ39    1538Ø          LD      HL,LINKS
3C92 3E3C      1539Ø          LD      A,LCOMM<-8       ;Establish 1st usable
3C94 77        154ØØ          LD      (HL),A           ;Init to 1st available
3C95 2C        1541Ø          INC     L                ;  page buffer
3C96 7Ø        1542Ø          LD      (HL),B           ;Init to highest page
3C97 2C        1543Ø          INC     L                ;  buffer available
3C98 77        1544Ø          LD      (HL),A           ;Init to begin & highest
3C99 2C        1545Ø          INC     L
3C9A 7Ø        1546Ø          LD      (HL),B
               1547Ø ;
               1548Ø ;        Establish page buffer linkage table
               1549Ø ;
3C9B 6F        155ØØ DOLINKS  LD      L,A              ;Init memory begin to
3C9C 3C        1551Ø          INC     A                ;  high bytes for as many
3C9D 77        1552Ø          LD      (HL),A           ;  bytes as pages to top
3C9E B8        1553Ø          CP      B
3C9F 2ØFA      1554Ø          JR      NZ,DOLINKS
```

COMM initialization code

```
3CA1 6F        15550          LD      L,A
3CA2 3600      15560          LD      (HL),0          ;Close out with zero
               15570 ;
               15580 ;        Establish starting page buffers for devices
               15590 ;
3CA4 2604      15600          LD      H,4             ;Init 1st at links+4
3CA6 DD219538  15610          LD      IX,KIVCTR
3CAA CDE63C    15620          CALL    INITBUF         ;Init *KI page buffer
3CAD DD219938  15630          LD      IX,PRVCTR
3CB1 CDE63C    15640          CALL    INITBUF         ;Init *PR page buffer
3CB4 DD219D38  15650          LD      IX,CLREC
3CB8 CDE63C    15660          CALL    INITBUF         ;Init *CL-R page buffer
3CBB DD21A138  15670          LD      IX,CLSEND
3CBF CDE63C    15680          CALL    INITBUF         ;Init *CL-S page buffer
3CC2 DD21A538  15690          LD      IX,FSVCTR
3CC6 CDE63C    15700          CALL    INITBUF         ;Init *FS page buffer
3CC9 DD21A938  15710          LD      IX,FRVCTR
3CCD CDE63C    15720          CALL    INITBUF         ;Init *FR page buffer
               15730 ;
               15740 ;        Calculate free buffer space
               15750 ;
3CD0 2639      15760          LD      H,LINKS<-8      ;P/u hi-order link table
3CD2 0600      15770          LD      B,0             ;Init count to zero
3CD4 3A0039    15780          LD      A,(LINKS)       ;Find pointer to 1st spr
3CD7 6F        15790          LD      L,A
3CD8 7E        15800 FBS1     LD      A,(HL)          ;P/u pointer to next
3CD9 B7        15810          OR      A               ;  spare & test if last
3CDA 2804      15820          JR      Z,FBS2          ;Exit if no more
3CDC 04        15830          INC     B               ;Bump counter
3CDD 6F        15840          LD      L,A             ;Show new pointer
3CDE 18F8      15850          JR      FBS1
3CE0 78        15860 FBS2     LD      A,B             ;Transfer the count
3CE1 32AD38    15870          LD      (FREEPG),A      ;  and save it
3CE4 1818      15880          JR      SETUPT
               15890 ;
               15900 ;        Routine to establish starting page buffers
               15910 ;
3CE6 DD360000  15920 INITBUF  LD      (IX),0          ;Show low-order PUT/GET
3CEA DD360200  15930          LD      (IX+2),0        ;  start at 0 reference
3CEE E5        15940          PUSH    HL
3CEF CD3635    15950          CALL    NEXTAP          ;Find next available page
3CF2 CA463D    15960          JP      Z,NOBUFS        ;Go if insufficient pages
3CF5 E1        15970          POP     HL
3CF6 DD7701    15980          LD      (IX+1),A        ;Set high-order PUT/GET
3CF9 DD7703    15990          LD      (IX+3),A        ;  page index pointers
3CFC 24        16000          INC     H               ;Bump to next entry in
3CFD C9        16010          RET                     ;  link table & return
               16020 ;
               16030 ;        Routine to set up the task processor
               16040 ;
               16050 SETUPT
               16060          IF      .NOT.BUFFRD
               16070          LD      DE,TCB8         ;CL task process
               16080          LD      C,8
3CFE           16090          @@ADTSK
               00180          LD      A,29
               00181          RST     40
               16100          LD      DE,TCB9         ;Printer output task
               16110          LD      C,9             ;Only if RS232 does
```

COMM initialization code

```
3CFE           1612Ø          @@ADTSK                     ;  not interrupt
               ØØ182          LD      A,29
               ØØ183          RST     4Ø
               1613Ø          ENDIF
               1614Ø ;
               1615Ø          IF      BUFFRD
3CFE 11E437    1616Ø          LD      DE,CLDCB            ;Turn on wakeup feature
3DØ1 FD21F833  1617Ø          LD      IY,TASK8A           ;Wakeup driver address
3DØ5 ØEØ4      1618Ø          LD      C,4                 ;Set addr CTL value
3DØ7 F3        1619Ø          DI
3DØ8           162ØØ          @@CTL                       ;Send to Com driver
3DØ8 3EØ5      ØØ184          LD      A,5
3DØA EF        ØØ185          RST     4Ø
3DØB FB        1621Ø          EI
3DØC FD221432  1622Ø          LD      (OLDVEC),IY         ;Save previous state
               1623Ø          ENDIF
               1624Ø ;
3D1Ø 21EE3D    1625Ø          LD      HL,LFEEDS           ;Clear most of screen
3D13           1626Ø          @@DSPLY
               ØØ186          IFEQ    ØØH,1
               ØØ187          LD      HL,
               ØØ188          ENDIF
3D13 3EØA      ØØ189          LD      A,1Ø
3D15 EF        ØØ19Ø          RST     4Ø
               1627Ø ;
               1628Ø ;        Transfer any translation characters
               1629Ø ;
3D16 3AEA3E    163ØØ          LD      A,(XLATES+1)        ;Transfer the output
3D19 32643Ø    1631Ø          LD      (XLTS1+1),A         ;  translation character
3D1C 3AE93E    1632Ø          LD      A,(XLATES)
3D1F 32683Ø    1633Ø          LD      (XLTS2+1),A
               1634Ø ;
3D22 3AEC3E    1635Ø          LD      A,(XLATER+1)        ;Transfer the input
3D25 322B34    1636Ø          LD      (XLTR1+1),A         ;  translation character
3D28 3AEB3E    1637Ø          LD      A,(XLATER)
3D2B 322F34    1638Ø          LD      (XLTR2+1),A
               1639Ø ;
3D2E 3AE33E    164ØØ          LD      A,(NULLPRM)         ;Transfer the null parm
3D31 323434    1641Ø          LD      (ACCNUL+1),A
3D34 3AE53E    1642Ø          LD      A,(XONP)            ;Transfer the XON/XOFF
3D37 32A93Ø    1643Ø          LD      (XONP1),A           ;  parms
3D3A 3AE73E    1644Ø          LD      A,(XOFFP)
3D3D 32AD3Ø    1645Ø          LD      (XOFFP1),A
3D4Ø 322F3Ø    1646Ø          LD      (XOFFP2),A
3D43 C3233Ø    1647Ø          JP      MAINLP
               1648Ø ;
               1649Ø ;        Error handling on initialization
               165ØØ ;
3D46 21413E    1651Ø NOBUFS   LD      HL,NOBUFS$          ;"Not enuf mem for buffers
3D49 DD        1652Ø          DB      ØDDH
3D4A 21223E    1653Ø BADCL    LD      HL,BADCL$           ;"Need RS-232 device name
3D4D           1654Ø          @@LOGOT
               ØØ191          IFEQ    ØØH,1
               ØØ192          LD      HL,
               ØØ193          ENDIF
3D4D 3EØC      ØØ194          LD      A,12
3D4F EF        ØØ195          RST     4Ø
3D5Ø C3ØA3Ø    1655Ø          JP      $ABORT
               1656Ø ;
```

COMM initialization code

```
                16570 ;         Messages
                16580 ;
3D53 43         16590 HELLO$    DB      'COMM'
     4F 4D 4D
3D57            16600 *GET      CLIENT:3
                16610 ;CLIENTS/ASM - File to establish sign-on headers
                16620 ;
3D57 2Ø         1663Ø           DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
3D81 2Ø         1664Ø           DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                1665Ø ;
3D96 41         1666Ø           DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
3DBE 2Ø         1667Ø           DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
                1668Ø           IF      @MOD4
3DD6 55         1669Ø GETMNU$ DB        'Use <CLEAR-8> for menu',LF,CR
     73 65 2Ø 3C 43 4C 45 41
     52 2D 38 3E 2Ø 66 6F 72
     2Ø 6D 65 6E 75 ØA ØD
                167ØØ           ENDIF
                1671Ø           IF      @MOD2
                1672Ø GETMNU$ DB        'Use <ESC-8> for menu',LF,CR
                1673Ø           ENDIF
3DEE ØA         1674Ø LFEEDS    DB      LF,LF,LF,LF,LF,LF,LF
     ØA ØA ØA ØA ØA ØA
3DF5 ØA         1675Ø           DB      LF,LF,LF,LF,LF,LF,LF,LF,LF,LF,LF,14,3
     ØA ØA ØA ØA ØA ØA ØA ØA
     ØA ØA ØE Ø3
                1676Ø ;
3EØ2 ØØ         1677Ø           DC      32,Ø            ;Patch space
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
                1678Ø ;
3E22 43         1679Ø BADCL$    DB      'Comm Line driver not specified',CR
     6F 6D 6D 2Ø 4C 69 6E 65
     2Ø 64 72 69 76 65 72 2Ø
     6E 6F 74 2Ø 73 7Ø 65 63
     69 66 69 65 64 ØD
3E41 49         168ØØ NOBUFS$ DB        'Insufficient memory to establish buffers',CR
     6E 73 75 66 66 69 63 69
     65 6E 74 2Ø 6D 65 6D 6F
     72 79 2Ø 74 6F 2Ø 65 73
     74 61 62 6C 69 73 68 2Ø
```

COMM initialization code

```
     62 75 66 66 65 72 73 ØD
3E6A 2Ø          1681Ø DEVICE$ DB       ' KI  PR CL-RCL-S FS  FR ????'
     4B 49 2Ø 2Ø 5Ø 52 2Ø 43
     4C 2D 52 43 4C 2D 53 2Ø
     46 53 2Ø 2Ø 46 52 2Ø 3F
     3F 3F 3F
3E86 2A          1682Ø OVRRUN$ DB       '** xxxx Buffer overrun **',3
     2A 2Ø 78 78 78 78 2Ø 42
     75 66 66 65 72 2Ø 6F 76
     65 72 72 75 6E 2Ø 2A 2A
     Ø3
3EAØ 5Ø          1683Ø PRNAME  DB       'PR'
     52
                 1684Ø ;
3EA2 58          1685Ø PRMTBL$ DB       'XLATES'
     4C 41 54 45 53
3EA8 E93E        1686Ø         DW       XLATES
3EAA 58          1687Ø         DB       'XS    '
     53 2Ø 2Ø 2Ø 2Ø
3EBØ E93E        1688Ø         DW       XLATES
3EB2 58          1689Ø         DB       'XLATER'
     4C 41 54 45 52
3EB8 EB3E        169ØØ         DW       XLATER
3EBA 58          1691Ø         DB       'XR    '
     52 2Ø 2Ø 2Ø 2Ø
3ECØ EB3E        1692Ø         DW       XLATER
3EC2 4E          1693Ø         DB       'NULL  '
     55 4C 4C 2Ø 2Ø
3EC8 E33E        1694Ø         DW       NULLPRM
3ECA 4E          1695Ø         DB       'N     '
     2Ø 2Ø 2Ø 2Ø 2Ø
3EDØ E33E        1696Ø         DW       NULLPRM
3ED2 58          1697Ø         DB       'XON   '
     4F 4E 2Ø 2Ø 2Ø
3ED8 E53E        1698Ø         DW       XONP
3EDA 58          1699Ø         DB       'XOFF  '
     4F 46 46 2Ø 2Ø
3EEØ E73E        17ØØØ         DW       XOFFP
3EE2 ØØ          17Ø1Ø         NOP
                 17Ø2Ø ;
3EE3 FFFF        17Ø3Ø NULLPRM DW       -1              ;Default to accept nulls
3EE5 11ØØ        17Ø4Ø XONP    DW       'Q'-4ØH         ;Ctl-Q
3EE7 13ØØ        17Ø5Ø XOFFP   DW       'S'-4ØH         ;Ctl-S
3EE9 ØØØØ        17Ø6Ø XLATES  DW       Ø
3EEB ØØØØ        17Ø7Ø XLATER  DW       Ø
                 17Ø8Ø ;
3EED             ØØ11Ø         SUBTTL   <>
3CØØ             ØØ12Ø         END      LCOMM
```

```
$ABORT          3ØØA $ERROR          3Ø1A $EXIT           3ØØØ
$OPEN           3ØØF @@1             ØØØØ @@2             ØØØØ
@@3             ØØØØ @@4             ØØØØ @MOD2           ØØØØ
@MOD4           FFFF ACCLFSW         3ØDA ACCNUL          3433
AUTXOFF         346A BADCL           3D4A BADCL$          3E22
BASE            3ØØØ BRAKET          35BE BREAK           ØØ8Ø
BUFFRD          FFFF BUFGET          3516 CFLAG           3582
CKFREPG         3Ø45 CLDCB           37E4 CLOUT           3ØFA
CLREC           389D CLS             328F CLSEND          38A1
CMDERR          327D CMDERR$         378D CMDKEY          3168
CMDPMT          359A CMPLTD          35AB CR              ØØØD
CRSW            3ØEØ CTLQ            3ØB2 CTLR            3ØC2
CVD1            333Ø CVD2            3335 DEVICE$         3E6A
DEVOUT          3116 DOLINKS         3C9B DOSCMD          3565
DPLXSW          3Ø69 DSPCTRL         3Ø88 DUMMY           3844
DUMPCHR         34F6 ECHOSW          3ØC8 ECLF1           31Ø9
ECOLF           31ØA EIGHT           341E ENUFPG          3Ø37
EOFFS           3Ø5C EXIT            32ØF FBS1            3CD8
FBS2            3CEØ FILEOF          338D FILEPMT         35C3
FILFR           33CC FILID           3396 FILR1           3355
FILR2           336Ø FILR3           3369 FILR4           336B
FILRES          334C FILREW          3384 FNDDVR          34ØØ
FNPIU           354E FRCIT           3467 FRCPUT          3451
FREEPG          38AD FRIOSW          3147 FRIO_OFF        33E4
FRNAME$         3773 FRSW            312D FRVCTR          38A9
FR_FCB          3824 FR_OFF          33DF FSNAME$         3769
FSOFF           3Ø77 FSSW            3Ø4Ø FSSWGO          3Ø5Ø
FSVCTR          38A5 FS_FCB          38Ø4 FS_OFF          33DA
GETMNU$         3DD6 GOTNAP          3544 HELLO$          3D53
HIPAGE          39Ø1 INITBUF         3CE6 INNEOUT         3527
ISBRK           3491 KEEPCH          3438 KISW            348Ø
KIVCTR          3895 LCMON           3Ø7Ø LCOMM           3CØØ
LCOMMA          3CØ9 LF              ØØØA LFEEDS          3DEE
LILPG$          37AC LINKS           39ØØ LNKD            341A
MAINLP          3Ø23 MENU            329A MNUMSG          35E5
NEXTAP          3536 NOBUFS          3D46 NOBUFS$         3E41
NOSQ            3ØB8 NOTBRK          3485 NOTCLS          3129
NOTCR           3ØDF NOTNOW          33D3 NULLPRM         3EE3
OFFS            337C OLDVEC          3214 OPENMSG         35D1
OUTPGM          38AE OUTPUT          34D6 OVRRUN$         3E86
PAGSPR$         377F PGMGET          38B4 POPERR          3267
PRDCB           34BD PRLOOP          34BA PRMTBL$         3EA2
PRNAME          3EAØ PRVCTR          3899 PUTPR           313A
QCL             3238 QFUNC           3233 QONOFF          325A
QONOFF1         325F QSHAKE          3242 QSHAKE1         3251
QUIT$           3ØØ3 RCVBUF          3BØØ RETADD          341D
SAVCHR          3ØA1 SENDIT          3Ø62 SETUPT          3CFE
SFLG            3Ø11 SHAKE           3ØA2 SKIPREC         3ØF4
SNDOUT          31Ø2 STACK           3ØØ4 STAT1           35E6
STAT2           37Ø5 STATAB          372D STATLP1         32B4
STATLP2         32D8 STATLP3         33Ø1 STATLP4         3324
TAKEREC         3ØF1 TASK8A          33F8 TASK8B          3444
TASK9           34B8 TASKK           3474 TASKS           33E9
TSTNUL          343Ø TURNOF          3284 TURNON          3286
XLATER          3EEB XLATES          3EE9 XLTR1           342A
XLTR2           342E XLTS1           3Ø63 XLTS2           3Ø67
XMTBUF          3AØØ XOFF            ØØ13 XOFFP           3EE7
XOFFP1          3ØAD XOFFP2          3Ø2F XONP            3EE5
```

```
XONP1           3ØA9 @@ABORT        6C16 @@ADTSK          6CA9
@@BANK          71C1 @@BKSP         6EA1 @@BREAK          71D7
@@CHNIO         6CØ1 @@CKBRKC       7225 @@CKDRV          6CFD
@@CKEOF         6EB6 @@CKTSK        6C94 @@CLOSE          6E8C
@@CLS           72ØF @@CMNDI        6C4Ø @@CMNDR          6C55
@@CTL           6A65 @@DATE         6BD7 @@DCSTAT         6D3C
@@DEBUG         6C7F @@DECHEX       7141 @@DIRRD          7ØAE
@@DIRWR         7ØC3 @@DIV16        712C @@DIV8           7117
@@DODIR         6D12 @@DSP          6A29 @@DSPLY          6AC9
@@ERROR         6C6A @@EXIT         6C2B @@FEXT           7Ø1B
@@FLAGS         71AB @@FNAME        7Ø3Ø @@FSPEC          7ØØ6
@@GATRD         7Ø99 @@GATWR        7ØD8 @@GET            6A3D
@@GTDCB         7Ø5A @@GTDCT        7Ø45 @@GTMOD          7Ø6F
@@HDFMT         6DE4 @@HEX16        718Ø @@HEX8           716B
@@HEXDEC        7156 @@HIGH$        7195 @@INIT           6E62
@@KBD           6AA1 @@KEY          6A15 @@KEYIN          6AB5
@@KLTSK         6CE8 @@LOAD         6FDC @@LOC            6ECB
@@LOF           6EEØ @@LOGER        6BØØ @@LOGOT          6B15
@@MSG           6B4C @@MUL16        71Ø2 @@MUL8           7ØED
@@OPEN          6E77 @@PARAM        6BC2 @@PAUSE          6BAD
@@PEOF          6EF5 @@POSN         6FØA @@PRINT          6B61
@@PRT           6A79 @@PUT          6A51 @@RAMDIR         6D27
@@RDSEC         6DBA @@RDSSC        7Ø84 @@READ           6F1F
@@REMOV         6E4D @@RENAM        6E38 @@REW            6F34
@@RMTSK         6CBE @@RPTSK        6CD3 @@RREAD          6F49
@@RSLCT         6DA5 @@RSTOR        6D66 @@RUN            6FF1
@@RWRIT         6F5E @@SEEK         6D9Ø @@SEEKSC         6F73
@@SKIP          6F88 @@SLCT         6D51 @@STEPI          6D7B
@@TIME          6BEC @@VDCTL        6B98 @@VER            6F9D
@@VRSEC         6DCF @@WEOF         6FB2 @@WHERE          6A8D
@@WRITE         6FC7 @@WRSEC        6DF9 @@WRSSC          6EØE
@@WRTRK         6E23
ØØØØØ Total errors
```

NOTES:

NOTES:

CONV/CMD - Convert Model III TRSDOS files

The Conv utility will move files from TRSDOS 1.2 and 1.3, Model III, copying them to an LS-DOS or TRSDOS 6 disk.

```
                ;CONV/ASM - Convert TRSDOS 1.2, 1.3 Disks
ØØØØ                    TITLE   <CONV - LS-DOS 6.2>
                ;
ØØ1C            HOME    EQU     1CH
ØØ1F            CLR     EQU     1FH
ØØØ3            ETX     EQU     Ø3H
ØØØD            CR      EQU     ØDH
ØØØA            LF      EQU     1Ø
                ;
ØØ4Ø            FLAG    EQU     Ø1ØØØØØØB
ØØ1Ø            ABB     EQU     ØØØ1ØØØØB
                ;
ØØØØ            *GET    SVCMAC:3                ;SVC Macro equivalents
                ;SVCMAC/ASM - LS-DOS Version VI
                *LIST   OFF
                *LIST   ON
ØØØØ            *GET    COPYCOM:3               ;Copyright message
                ; COPYCOM - File for Copyright COMment block
                ;
ØØØØ                    COM     '<*(C) 1982,83,84 by LSI*>'
                ;
26ØØ                    ORG     26ØØH
                ;
                BEGIN
26ØØ                    @@CKBRKC
26ØØ 3E6A               LD      A,1Ø6
26Ø2 EF                 RST     4Ø
26Ø3 28Ø4               JR      Z,BEGINA        ;Continue if no break
26Ø5 21FFFF             LD      HL,-1
26Ø8 C9                 RET                     ;  else abort
                ;
                BEGINA
26Ø9 ED734626           LD      (STACK),SP      ;Save entry stack
26ØD E5                 PUSH    HL              ;Save ptr to CMD buffer
26ØE                    @@DSPLY HELLO$          ;Display the signon
                        IFEQ    Ø1H,1
26ØE 21C32A             LD      HL,HELLO$
                        ENDIF
2611 3EØA               LD      A,1Ø
2613 EF                 RST     4Ø
2614 21ØØØØ             LD      HL,Ø            ;Set up to get HIGH$
2617 45                 LD      B,L
2618                    @@FLAGS                 ;IY => flag table base
2618 3E65               LD      A,1Ø1
261A EF                 RST     4Ø
261B FDCBØ24E           BIT     1,(IY+'C'-'A')  ;OK if not CMDR
261F 28Ø1               JR      Z,NOTCMDR       ;Use LOW$ otherwise
2621 Ø4                 INC     B
2622            NOTCMDR @@HIGH$                 ;P/u HIGH$/LOW$
2622 3E64               LD      A,1ØØ
2624 EF                 RST     4Ø
2625 22D528             LD      (MYHIGH),HL     ;Store away
2628 FDE5               PUSH    IY              ;Trans to HL
262A D1                 POP     DE
262B 21ØAØØ             LD      HL,'K'-'A'      ;Offset to KFLAG$
262E 19                 ADD     HL,DE           ;HL=>KFLAG$
262F 222427             LD      (KFLG),HL       ;Store pointer
2632 CB86               RES     Ø,(HL)          ;Kick break bit off
2634 2112ØØ             LD      HL,'S'-'A'      ;SFLAG$ offset
2637 19                 ADD     HL,DE
2638 22E427             LD      (SFLG),HL       ;Store away
```

```
263B E1         ØØ54Ø          POP     HL              ;Restore cmd pointer
263C CD5526     ØØ55Ø          CALL    PGRM            ;  and continue
                ØØ56Ø ;
                ØØ57Ø ;        Exit routines
                ØØ58Ø ;
263F 21ØØØØ     ØØ59Ø $EXIT    LD      HL,Ø            ;Init to no error
2642            ØØ6ØØ $QUIT    @@CKBRKC                ;Clear out break bit
2642 3E6A       ØØØ12          LD      A,1Ø6
2644 EF         ØØØ13          RST     4Ø
2645 31ØØØØ     ØØ61Ø          LD      SP,$-$          ;P/u original stack
2646            ØØ62Ø STACK    EQU     $-2
2648 C9         ØØ63Ø          RET
                ØØ64Ø ;
2649 21FFFF     ØØ65Ø $ABORT   LD      HL,-1           ;Set abort code
264C 18F4       ØØ66Ø          JR      $QUIT           ;  and quit
                ØØ67Ø ;
264E C5         ØØ68Ø $DSP     PUSH    BC              ;Display a character,
264F 4F         ØØ69Ø          LD      C,A             ;  saving BC
265Ø            ØØ7ØØ          @@DSP
265Ø 3EØ2       ØØØ14          LD      A,2
2652 EF         ØØØ15          RST     4Ø
2653 C1         ØØ71Ø          POP     BC
2654 C9         ØØ72Ø          RET
                ØØ73Ø ;
                ØØ74Ø ;        Pick up drive numbers and partial filespec
                ØØ75Ø ;
                ØØ76Ø PGRM:
2655 7E         ØØ77Ø          LD      A,(HL)          ;Check for NOT filespec
2656 FE2D       ØØ78Ø          CP      '-'             ;  char used
2658 2ØØ6       ØØ79Ø          JR      NZ,MVNAM1       ;Go if not NOT
265A 3EFF       ØØ8ØØ          LD      A,ØFFH          ;TRUE value
265C 32B72A     ØØ81Ø          LD      (NOTPRM),A      ;Set if specified
265F 23         ØØ82Ø          INC     HL
266Ø 11B82A     ØØ83Ø MVNAM1   LD      DE,PATTRN       ;Point to possible partspec
2663 Ø6Ø8       ØØ84Ø          LD      B,8             ;Max 8 chars in name
2665 CDCF29     ØØ85Ø          CALL    SKIPSP          ;Skip spaces
2668 CDDE29     ØØ86Ø          CALL    MOVELT          ;Move letters/digits/$
266B CDD629     ØØ87Ø          CALL    SKIPLT          ;Skip letters/digits/$
266E 7E         ØØ88Ø          LD      A,(HL)          ;Check for extension
266F FE2F       ØØ89Ø          CP      '/'
2671 2ØØC       ØØ9ØØ          JR      NZ,NOEXT        ;Go if none
2673 23         ØØ91Ø          INC     HL
2674 11CØ2A     ØØ92Ø          LD      DE,PATEXT       ;Point to ext field
2677 Ø6Ø3       ØØ93Ø          LD      B,3             ;Max 3 chars in ext
2679 CDDE29     ØØ94Ø          CALL    MOVELT          ;Move letters/digits/$
267C CDD629     ØØ95Ø          CALL    SKIPLT          ;Skip letters/digits/$
267F CDBC29     ØØ96Ø NOEXT    CALL    GETDRV          ;Get source drive #
2682 32492C     ØØ97Ø          LD      (SDRIVE),A      ;Store drive #
2685 A7         ØØ98Ø          AND     A               ;Be sure not drive Ø
2686 11712B     ØØ99Ø          LD      DE,NOTØ         ;Error msg
2689 EB         Ø1ØØØ          EX      DE,HL
268A CA5529     Ø1Ø1Ø          JP      Z,PERR1         ;Param error source is Ø
268D EB         Ø1Ø2Ø          EX      DE,HL           ;Restore cmd line ptr
268E CDCF29     Ø1Ø3Ø          CALL    SKIPSP          ;Skip spaces
2691 CDB629     Ø1Ø4Ø          CALL    GETDRV2         ;Get destination drive
2694 324A2C     Ø1Ø5Ø          LD      (DDRIVE),A      ;ØFFH if no dest drv
2697 CDCF29     Ø1Ø6Ø          CALL    SKIPSP          ;Move to '('
                Ø1Ø7Ø ;
                Ø1Ø8Ø ;        Scan parameters
                Ø1Ø9Ø ;
269A 11822A     Ø11ØØ          LD      DE,PRMTBL$      ;Check parameters entered
```

```
269D                  @@PARAM
269D 3E11                    LD      A,17
269F EF                      RST     4Ø
26AØ C25129                  JP      NZ,PRMERR        ;Quit on parm error
26A3 21ØØØØ           DPARM  LD      HL,$-$           ;DIR only?
26A6 7C                      LD      A,H
26A7 B5                      OR      L
26A8 28Ø5                    JR      Z,SPARM          ;Go if not
26AA 3EFF                    LD      A,ØFFH           ;Set flag at DDRIVE
26AC 324A2C                  LD      (DDRIVE),A       ;If dest is ff, read DIR
26AF 21ØØØØ           SPARM  LD      HL,$-$           ;Check if no parms S,I,V
26B2 11ØØØØ           VPARM  LD      DE,$-$
26B5 Ø1ØØØØ           IPARM  LD      BC,$-$
26B8 7D                      LD      A,L
26B9 B3                      OR      E
26BA B1                      OR      C
26BB 323B27                  LD      (SIV+1),A        ;Save S!I!V
26BE 21FFFF           QPARM  LD      HL,ØFFFFH        ;P/u Q,N,O parms
26C1 11ØØØØ           NPARM  LD      DE,Ø
26C4 Ø1ØØØØ           OPARM  LD      BC,Ø
26C7 7B                      LD      A,E              ;Form N!O
26C8 B1                      OR      C
26C9 32F527                  LD      (NORO+1),A       ;Save that
                      ;
                      ;      Save old DCT
                      ;
26CC 3A492C                  LD      A,(SDRIVE)       ;Pick up source drive #
26CF 4F                      LD      C,A              ;Move to C reg
26DØ 3A4A2C                  LD      A,(DDRIVE)       ;Be sure not single drive
26D3 B9                      CP      C
26D4 21462B                  LD      HL,NOTONE        ;=>error msg
26D7 CA5529                  JP      Z,PERR1          ;Go if same
26DA                  @@GTDCT                         ;Point to DCT
26DA 3E51                    LD      A,81
26DC EF                      RST     4Ø
26DD C5                      PUSH    BC               ;Save drive #
26DE FDE5                    PUSH    IY               ;Move DCT to HL reg
26EØ E1                      POP     HL
26E1 11522C                  LD      DE,SAVDCT        ;Point to save area
26E4 Ø1ØAØØ                  LD      BC,1Ø
26E7 EDBØ                    LDIR                     ;Move it
26E9 C1                      POP     BC
                      ;
                      ;      Find directory track
                      ;
26EA 11Ø1ØØ                  LD      DE,ØØØ1          ;Track Ø, sector 1
26ED 21ØØ2D                  LD      HL,DBUFF         ;Buffer for sector
26FØ                  @@RDSEC
26FØ 3E31                    LD      A,49
26F2 EF                      RST     4Ø
26F3 28Ø8                    JR      Z,OKØ            ;Go if no error
26F5 FEØ6                    CP      6                ;Was it DAM error?
26F7 C23629                  JP      NZ,IOERR         ;Go if some other
26FA CD7Ø2A                  CALL    CKEARLY          ;Can we do this type?
26FD 23               OKØ    INC     HL               ;Point to dir cyl #
26FE 56                      LD      D,(HL)           ;Get it
26FF 24                      INC     H                ;Point to TRSDOS
27ØØ 2B                      DEC     HL               ;  version number
27Ø1 2B                      DEC     HL
27Ø2 2B                      DEC     HL
27Ø3 7E                      LD      A,(HL)           ;Pick it up
```

```
27Ø4 329C28              LD      (TRSDOS+1),A      ;Save for later
                 ;
                 ;       Read directory records into memory
                 ;
27Ø7 1EØ3                LD      E,3               ;Skip GAT and HIT
27Ø9 Ø61Ø                LD      B,16              ;Read 16 sectors
27ØB 21ØØ2D              LD      HL,DBUFF
27ØE FD36Ø712    DREAD   LD      (IY+7),18         ;Chg # sectors/trk for
2712                     @@RDSEC                   ;  TRSDOS & Read a sector
2712 3E31                LD      A,49
2714 EF                  RST     4Ø
2715 28Ø5                JR      Z,OK1             ;Go if no error
2717 FEØ6                CP      6                 ;Ignore record type
2719 C23629              JP      NZ,IOERR          ;Go if error
271C 24          OK1     INC     H                 ;Bump buffer pointer
271D 1C                  INC     E                 ;Bump sector number
271E 1ØEE                DJNZ    DREAD             ;Loop till done
                 ;
                 ;       Loop through all entries
                 ;
272Ø 21ØØ2D              LD      HL,DBUFF          ;Point to first entry
2723             ELOOP   EQU     $
2723 3AØØØØ              LD      A,($-$)           ;Check system break bit
2724             KFLG    EQU     $-2               ;Address of KFLAG
2726 CB47                BIT     Ø,A
2728 C24926              JP      NZ,$ABORT         ;Abort if set
272B 46                  LD      B,(HL)            ;P/U attributes
272C E5                  PUSH    HL
272D DDE1                POP     IX
272F E5                  PUSH    HL
273Ø CB6Ø                BIT     4,B               ;Alive?
2732 CAØF29              JP      Z,SKIPIT          ;Skip it if dead
2735 CB78                BIT     7,B               ;FXDE?
2737 C2ØF29              JP      NZ,SKIPIT         ;Skip it if so
                 ;
                 ;       Check file's attributes
                 ;
273A 3EØØ        SIV     LD      A,$-$             ;S, I, or V given?
273C A7                  AND     A
273D 2821                JR      Z,NOSIV           ;Go if none given
273F CB7Ø                BIT     6,B               ;SYS file?
2741 28Ø9                JR      Z,NOTSYS          ;Go if not
2743 3ABØ26              LD      A,(SPARM+1)       ;S parm given?
2746 A7                  AND     A
2747 CAØF29              JP      Z,SKIPIT          ;Skip file if not
274A 1814                JR      NOSIV             ;  else possible match
274C CB58        NOTSYS  BIT     3,B               ;Visible or invisible?
274E 2ØØ9                JR      NZ,INV            ;Go if inv
275Ø 3AB326              LD      A,(VPARM+1)       ;V parm given?
2753 A7                  AND     A
2754 CAØF29              JP      Z,SKIPIT          ;Skip file if not
2757 18Ø7                JR      NOSIV             ;  else possible match
2759 3AB626      INV     LD      A,(IPARM+1)       ;I parm given?
275C A7                  AND     A
275D CAØF29              JP      Z,SKIPIT          ;Skip file if not
                 ;
                 ;       Check if name matches wildcard
                 ;
276Ø 11Ø5ØØ      NOSIV   LD      DE,5              ;Offset to name field
2763 19                  ADD     HL,DE
2764 E5                  PUSH    HL                ;Compare with pattern
```

```
2765 11B82A            LD      DE,PATTRN       ;  of user partspec
2768 Ø6ØB              LD      B,11
276A 1A        CPLOOP  LD      A,(DE)          ;P/U pattern byte
276B 13                INC     DE
276C FE24              CP      '$'             ;Matchall?
276E 28Ø3              JR      Z,MATCH
277Ø BE                CP      (HL)            ;Match?
2771 2ØØ3              JR      NZ,NMATCH       ;Go if not
2773 23        MATCH   INC     HL
2774 1ØF4              DJNZ    CPLOOP
2776 E1        NMATCH  POP     HL              ;Z if match, NZ if not
2777 CDØB2A            CALL    NOTCHK          ;Reverse flag if NOT entered
277A C2ØF29            JP      NZ,SKIPIT       ;Skip file if no match
               ;
277D 11Ø92C            LD      DE,FCB          ;Point to FCB
278Ø Ø6Ø8              LD      B,8
2782 7E        MVNAME  LD      A,(HL)          ;Move name
2783 FE2Ø              CP      ' '             ;Space?
2785 28Ø5              JR      Z,GOTNAM        ;Go if hit one
2787 23                INC     HL
2788 12                LD      (DE),A          ;Put to FCB
2789 13                INC     DE
278A 1ØF6              DJNZ    MVNAME
278C 48        GOTNAM  LD      C,B             ;Offset to ext field
278D Ø6ØØ              LD      B,Ø
278F Ø9                ADD     HL,BC
279Ø 7E                LD      A,(HL)          ;No extension?
2791 FE2Ø              CP      ' '
2793 281Ø              JR      Z,GOTEXT        ;Go if so
2795 3E2F              LD      A,'/'           ;Put in slash
2797 12                LD      (DE),A
2798 13                INC     DE
2799 Ø6Ø3              LD      B,3
279B 7E        EXLOOP  LD      A,(HL)          ;Move extension
279C 23                INC     HL
279D FE2Ø              CP      ' '             ;Finished?
279F 28Ø4              JR      Z,GOTEXT
27A1 12                LD      (DE),A
27A2 13                INC     DE
27A3 1ØF6              DJNZ    EXLOOP          ;Loop till done
               ;
27A5 3EØ3      GOTEXT  LD      A,ETX           ;Put ETX at end for dsply
27A7 12                LD      (DE),A
27A8 D5                PUSH    DE              ;Save current spot in FCB
27A9 21Ø92C            LD      HL,FCB          ;Move name to buffer
27AC 11E92B            LD      DE,FNAME        ;  for printing
27AF Ø12ØØØ            LD      BC,32
27B2 EDBØ              LDIR
27B4 D1                POP     DE              ;Get back where we were
               ;
               ;Print filenames if no destination drive (DDRIVE=ØFFH)
               ;
27B5 3A4A2C            LD      A,(DDRIVE)      ;Check for just printing DIR
27B8 3C                INC     A               ;Set Z if FF
27B9 2ØØ6              JR      NZ,MOVING       ;Go if not FF
27BB CD1C2A            CALL    SHOW            ;Print entry
27BE C3ØF29            JP      SKIPIT          ;  and go on to next
               ;
               ;       Check if file exists on destination disk
               ;
27C1 3E3A      MOVING  LD      A,':'           ;Now put the drive separator
```

```
27C3 12         Ø286Ø         LD      (DE),A          ;  in the FCB
27C4 13         Ø287Ø         INC     DE
27C5 3A4A2C     Ø288Ø         LD      A,(DDRIVE)      ;Put in drive spec
27C8 F63Ø       Ø289Ø         OR      'Ø'             ;Change number to ASCII
27CA 12         Ø29ØØ         LD      (DE),A
27CB 13         Ø291Ø         INC     DE
27CC 3EØ3       Ø292Ø         LD      A,ETX           ;Put in ETX to end
27CE 12         Ø293Ø         LD      (DE),A
27CF 21Ø92C     Ø294Ø         LD      HL,FCB          ;Copy into 2nd FCB
27D2 11292C     Ø295Ø         LD      DE,FCB2
27D5 Ø12ØØØ     Ø296Ø         LD      BC,32
27D8 EDBØ       Ø297Ø         LDIR
27DA 11292C     Ø298Ø         LD      DE,FCB2         ;Point to start of FCB
27DD 21ØØ3D     Ø299Ø         LD      HL,TBUFF        ;Point to transfer buffer
27EØ Ø6ØØ       Ø3ØØØ         LD      B,Ø             ;LRL=256
27E2 E5         Ø3Ø1Ø         PUSH    HL
27E3 21ØØØØ     Ø3Ø2Ø         LD      HL,$-$          ;HL => SFLAG
27E4            Ø3Ø3Ø SFLG    EQU     $-2
27E6 CBC6       Ø3Ø4Ø         SET     Ø,(HL)          ;Set the open inhibit bit
27E8 E1         Ø3Ø5Ø         POP     HL
27E9            Ø3Ø6Ø         @@OPEN                  ;Do the open
27E9 3E3B       ØØØ24         LD      A,59
27EB EF         ØØØ25         RST     4Ø
27EC 47         Ø3Ø7Ø         LD      B,A             ;Save return code
27ED 28Ø5       Ø3Ø8Ø         JR      Z,NORO          ;Go if opened okay
27EF FE18       Ø3Ø9Ø         CP      18H             ;File not found?
27F1 C23629     Ø31ØØ         JP      NZ,IOERR        ;  else an error
                Ø311Ø ;
                Ø312Ø ;       Check New and Old parms
                Ø313Ø ;
27F4 3EØØ       Ø314Ø NORO    LD      A,Ø             ;N or O specified?
27F6 A7         Ø315Ø         AND     A
27F7 2816       Ø316Ø         JR      Z,CHECKQ        ;Go if neither
27F9 3AC526     Ø317Ø         LD      A,(OPARM+1)     ;O parm given?
27FC A7         Ø318Ø         AND     A
27FD 28Ø4       Ø319Ø         JR      Z,CKNEW         ;Go if not
27FF AF         Ø32ØØ         XOR     A
28ØØ BØ         Ø321Ø         OR      B               ;Did file exist?
28Ø1 28ØC       Ø322Ø         JR      Z,CHECKQ        ;Go if so (ok)
28Ø3 3AC226     Ø323Ø CKNEW   LD      A,(NPARM+1)     ;N parm given?
28Ø6 A7         Ø324Ø         AND     A
28Ø7 CAØF29     Ø325Ø         JP      Z,SKIPIT        ;Skip file if not
28ØA AF         Ø326Ø         XOR     A
28ØB BØ         Ø327Ø         OR      B               ;Be sure it was new
28ØC CAØF29     Ø328Ø         JP      Z,SKIPIT        ;Go if it wasn't
                Ø329Ø ;
                Ø33ØØ ;       Ask question if Q parm was given
                Ø331Ø ;
28ØF 3ABF26     Ø332Ø CHECKQ  LD      A,(QPARM+1)     ;Check Q parm
2812 A7         Ø333Ø         AND     A
2813 2Ø13       Ø334Ø         JR      NZ,QUERY        ;Query if so
2815 21CA2B     Ø335Ø         LD      HL,CONVS        ;"Converting..."
2818            Ø336Ø         @@DSPLY
                ØØØ26         IFEQ    ØØH,1
                ØØØ27         LD      HL,
                ØØØ28         ENDIF
2818 3EØA       ØØØ29         LD      A,1Ø
281A EF         ØØØ3Ø         RST     4Ø
281B 21E92B     Ø337Ø         LD      HL,FNAME        ;Filename
281E            Ø338Ø         @@DSPLY
                ØØØ31         IFEQ    ØØH,1
```

```
               ØØØ32          LD      HL,
               ØØØ33          ENDIF
281E 3EØA      ØØØ34          LD      A,1Ø
282Ø EF        ØØØ35          RST     4Ø
2821 3EØD      Ø339Ø          LD      A,CR             ;Carriage return
2823 CD4E26    Ø34ØØ          CALL    $DSP
2826 1841      Ø341Ø          JR      TAKEIT1          ;Go & move it
               Ø342Ø ;
2828 21DC2B    Ø343Ø QUERY    LD      HL,CONVQ         ;"Convert file
282B           Ø344Ø          @@DSPLY                  ;Display it
               ØØØ36          IFEQ    ØØH,1
               ØØØ37          LD      HL,
               ØØØ38          ENDIF
282B 3EØA      ØØØ39          LD      A,1Ø
282D EF        ØØØ4Ø          RST     4Ø
282E 21A92B    Ø345Ø          LD      HL,QMARK         ;"?"
2831           Ø346Ø          @@DSPLY
               ØØØ41          IFEQ    ØØH,1
               ØØØ42          LD      HL,
               ØØØ43          ENDIF
2831 3EØA      ØØØ44          LD      A,1Ø
2833 EF        ØØØ45          RST     4Ø
2834 214D2C    Ø347Ø          LD      HL,ABUFF         ;Get answer
2837 Ø1ØØØ3    Ø348Ø          LD      BC,3<8           ;3 char max
283A           Ø349Ø          @@KEYIN
283A 3EØ9      ØØØ46          LD      A,9
283C EF        ØØØ47          RST     4Ø
283D DA4926    Ø35ØØ          JP      C,$ABORT         ;Abort if BREAK hit
284Ø 7E        Ø351Ø          LD      A,(HL)           ;Check for 'Y'
2841 CBAF      Ø352Ø          RES     5,A              ;Force upper case
2843 FE59      Ø353Ø          CP      'Y'
2845 C2ØF29    Ø354Ø          JP      NZ,SKIPIT        ;Skip it if not 'Y'
               Ø355Ø ;
               Ø356Ø ;        If file exists, query user
               Ø357Ø ;
2848 3A292C    Ø358Ø          LD      A,(FCB2)         ;Was file opened ok?
284B CB7F      Ø359Ø          BIT     7,A              ;Z = not found
284D 281A      Ø36ØØ          JR      Z,TAKEIT1        ;Go if it does not exist
284F 21AC2B    Ø361Ø          LD      HL,EXISTQ        ;"File exists, replace?
2852           Ø362Ø          @@DSPLY                  ;Print question
               ØØØ48          IFEQ    ØØH,1
               ØØØ49          LD      HL,
               ØØØ5Ø          ENDIF
2852 3EØA      ØØØ51          LD      A,1Ø
2854 EF        ØØØ52          RST     4Ø
2855 214D2C    Ø363Ø          LD      HL,ABUFF
2858 Ø1ØØØ3    Ø364Ø          LD      BC,3<8
285B           Ø365Ø          @@KEYIN                  ;Get answer
285B 3EØ9      ØØØ53          LD      A,9
285D EF        ØØØ54          RST     4Ø
285E DA4926    Ø366Ø          JP      C,$ABORT         ;Abort if break
2861 7E        Ø367Ø          LD      A,(HL)           ;Check answer
2862 CBAF      Ø368Ø          RES     5,A              ;Force uppercase
2864 FE59      Ø369Ø          CP      'Y'
2866 C2ØF29    Ø37ØØ          JP      NZ,SKIPIT        ;Skip if 'no'
               Ø371Ø ;
               Ø372Ø ;        Init file if it didn't exist
               Ø373Ø ;
2869 11292C    Ø374Ø TAKEIT1  LD      DE,FCB2
286C 1A        Ø375Ø          LD      A,(DE)           ;Was file opened?
286D CB7F      Ø376Ø          BIT     7,A              ;Z = not opened
```

```
286F 28Ø3      Ø377Ø          JR      Z,$+5           ;Remove existing file
2871           Ø378Ø          @@REMOV                 ;  for new LRL
2871 3E39      ØØØ55          LD      A,57
2873 EF        ØØØ56          RST     4Ø
2874 11Ø92C    Ø379Ø          LD      DE,FCB          ;Use other FCB now
2877 21ØØ3D    Ø38ØØ          LD      HL,TBUFF        ;Create file
287A DD46Ø4    Ø381Ø          LD      B,(IX+4)        ;P/U Mod III LRL
287D           Ø382Ø          @@INIT                  ;Create the file
287D 3E3A      ØØØ57          LD      A,58
287F EF        ØØØ58          RST     4Ø
288Ø C23629    Ø383Ø          JP      NZ,IOERR        ;Go if error
2883 D5        Ø384Ø          PUSH    DE              ;Change LRL to Ø for copy
2884 DDE3      Ø385Ø          EX      (SP),IX         ;IX to FCB start
2886 DDCBØ1BE  Ø386Ø          RES     7,(IX+1)        ;Show full sector ops
288A DD36Ø9ØØ  Ø387Ø          LD      (IX+9),Ø        ;Show LRL=Ø
288E DDE3      Ø388Ø          EX      (SP),IX         ;Switch back
289Ø D1        Ø389Ø          POP     DE
               Ø39ØØ ;
               Ø391Ø ;        Initialize to read from source file
               Ø392Ø ;
2891 E1        Ø393Ø TAKEIT2  POP     HL              ;Point to dir entry
2892 E5        Ø394Ø          PUSH    HL
2893 1114ØØ    Ø395Ø          LD      DE,2Ø           ;Point to ERN
2896 19        Ø396Ø          ADD     HL,DE
2897 5E        Ø397Ø          LD      E,(HL)          ;P/U ERN
2898 23        Ø398Ø          INC     HL
2899 56        Ø399Ø          LD      D,(HL)
289A 23        Ø4ØØØ          INC     HL              ;Leave ptg to extents
289B 3EØØ      Ø4Ø1Ø TRSDOS   LD      A,Ø             ;1.3 or later?
289D FE13      Ø4Ø2Ø          CP      13H
289F 38Ø7      Ø4Ø3Ø          JR      C,EARLY         ;Go if earlier than 1.3
28A1 DD7EØ3    Ø4Ø4Ø          LD      A,(IX+3)        ;Pick up EOF offset
28A4 A7        Ø4Ø5Ø          AND     A               ;Zero?
28A5 28Ø1      Ø4Ø6Ø          JR      Z,EARLY         ;No adjustment if so
28A7 13        Ø4Ø7Ø          INC     DE              ;If nonzero, adjust ERN
28A8 Ø6ØØ      Ø4Ø8Ø EARLY    LD      B,Ø             ;# sectors left in extent
28AA D5        Ø4Ø9Ø          PUSH    DE              ;Save ERN
28AB D9        Ø41ØØ          EXX                     ;Switch to alternate regs
               Ø411Ø ;
               Ø412Ø ;        Preallocate file
               Ø413Ø ;
28AC C1        Ø414Ø          POP     BC
28AD 78        Ø415Ø          LD      A,B             ;Empty file?
28AE B1        Ø416Ø          OR      C
28AF 281E      Ø417Ø          JR      Z,READ          ;Go if so
28B1 ØB        Ø418Ø          DEC     BC
28B2 11Ø92C    Ø419Ø          LD      DE,FCB          ;Point to FCB
28B5           Ø42ØØ          @@POSN                  ;Position to last sector
28B5 3E42      ØØØ59          LD      A,66
28B7 EF        ØØØ6Ø          RST     4Ø
28B8 28Ø9      Ø421Ø          JR      Z,OK3
28BA FE1C      Ø422Ø          CP      1CH             ;Ignore EOF errors
28BC 28Ø5      Ø423Ø          JR      Z,OK3
28BE FE1D      Ø424Ø          CP      1DH             ;  or past end errors
28CØ C23629    Ø425Ø          JP      NZ,IOERR        ;Quit on any others
28C3           Ø426Ø OK3      @@WRITE                 ;Write it
28C3 3E4B      ØØØ61          LD      A,75
28C5 EF        ØØØ62          RST     4Ø
28C6 C23629    Ø427Ø          JP      NZ,IOERR        ;Quit on write error
28C9           Ø428Ø          @@REW                   ;Position to start
28C9 3E44      ØØØ63          LD      A,68
```

```
28CB EF         ØØØ64          RST      4Ø
28CC C23629     Ø429Ø          JP       NZ,IOERR
                Ø43ØØ ;
                Ø431Ø ;        Read sectors
                Ø432Ø ;
28CF Ø6ØØ       Ø433Ø READ     LD       B,Ø             ;Count sectors read
28D1 21ØØ3D     Ø434Ø          LD       HL,TBUFF        ;Point to transfer buffer
28D4 11ØØØØ     Ø435Ø          LD       DE,$-$
28D5            Ø436Ø MYHIGH   EQU      $-2             ;Stuff HIGH$ value
28D7 15         Ø437Ø          DEC      D               ;256 bytes back
28D8 CD5B29     Ø438Ø GETONE   CALL     GETSEC          ;Get next sector
28DB 2ØØB       Ø439Ø          JR       NZ,WRITE        ;Go if EOF
28DD Ø4         Ø44ØØ          INC      B               ;Count sector
28DE 24         Ø441Ø          INC      H               ;Point to next spot
28DF CDØ52A     Ø442Ø          CALL     CPHLDE          ;Compare HL and DE
28E2 3EØØ       Ø443Ø          LD       A,Ø             ;No error code
28E4 3ØØ2       Ø444Ø          JR       NC,WRITE        ;Go if mem full
28E6 18FØ       Ø445Ø          JR       GETONE          ;  else loop for more
                Ø446Ø ;
                Ø447Ø ;        Write sectors to destination file
                Ø448Ø ;
28E8 F5         Ø449Ø WRITE    PUSH     AF              ;Save completion type
28E9 11Ø92C     Ø45ØØ          LD       DE,FCB          ;Point to file fcb
28EC 21ØØ3D     Ø451Ø          LD       HL,TBUFF        ;Point to transfer buffer
28EF 22ØC2C     Ø452Ø WRLOOP   LD       (FCB+3),HL      ;Point FCB to buffer
28F2 78         Ø453Ø          LD       A,B             ;Zero to write?
28F3 A7         Ø454Ø          AND      A
28F4 28Ø9       Ø455Ø          JR       Z,WRDUN         ;Go if so
28F6            Ø456Ø          @@WRITE                  ;Write to file
28F6 3E4B       ØØØ65          LD       A,75
28F8 EF         ØØØ66          RST      4Ø
28F9 C23629     Ø457Ø          JP       NZ,IOERR        ;Quit on write error
28FC 24         Ø458Ø          INC      H
28FD 1ØFØ       Ø459Ø          DJNZ     WRLOOP          ;Loop till done
                Ø46ØØ ;
                Ø461Ø ;        Were we at EOF?
                Ø462Ø ;
28FF F1         Ø463Ø WRDUN    POP      AF              ;Restore completion type
29ØØ A7         Ø464Ø          AND      A               ;At end of file?
29Ø1 28CC       Ø465Ø          JR       Z,READ          ;Go if not
                Ø466Ø ;
                Ø467Ø ;        Copy over EOF offset
                Ø468Ø ;
29Ø3 DD7EØ3     Ø469Ø          LD       A,(IX+3)        ;P/U offset from dir
29Ø6 32112C     Ø47ØØ          LD       (FCB+8),A       ;Put into FCB
29Ø9            Ø471Ø          @@CLOSE                  ;  and close the file
29Ø9 3E3C       ØØØ67          LD       A,6Ø
29ØB EF         ØØØ68          RST      4Ø
29ØC C23629     Ø472Ø          JP       NZ,IOERR        ;Quit on close error
                Ø473Ø ;
                Ø474Ø ;        Increment to next entry and loop if not done
                Ø475Ø ;
29ØF E1         Ø476Ø SKIPIT   POP      HL
291Ø 113ØØØ     Ø477Ø          LD       DE,48           ;48 bytes per entry
2913 19         Ø478Ø          ADD      HL,DE
2914 7D         Ø479Ø          LD       A,L             ;End of sector?
2915 FEFØ       Ø48ØØ          CP       ØFØH
2917 2ØØ3       Ø481Ø          JR       NZ,NOTEOS       ;Go if not
2919 24         Ø482Ø          INC      H
291A 2EØØ       Ø483Ø          LD       L,Ø
291C 11ØØ3D     Ø484Ø NOTEOS   LD       DE,TBUFF        ;Done?
```

```
291F CDØ52A               CALL    CPHLDE          ;CP HL,DE
2922 DA2327               JP      C,ELOOP         ;Loop back if not done
                   ;
                   ;      Finished
                   ;
2925 3EØD                 LD      A,CR            ;Blank line
2927 CD4E26               CALL    $DSP
292A CD4529               CALL    BYEBYE          ;Restore DCT
292D C33F26               JP      $EXIT
                   ;
293Ø CD4529        QUIT   CALL    BYEBYE          ;Restore DCT
2933 C34926               JP      $ABORT
                   ;
                   ;      Error routines
                   ;
2936 CD4529        IOERR  CALL    BYEBYE          ;Restore DCT
2939 6F            IOERR1 LD      L,A             ;Entry from PRMERR
293A 26ØØ                 LD      H,Ø
293C F6CØ                 OR      ØCØH            ;Abbrev, return
293E 4F                   LD      C,A             ;Error code to C
293F                      @@ERROR                 ;  for error display
293F 3E1A                 LD      A,26
2941 EF                   RST     4Ø
2942 C34226               JP      $QUIT
                   ;
2945 FDE5          BYEBYE PUSH    IY              ;Move back DCT
2947 D1                   POP     DE
2948 21522C               LD      HL,SAVDCT       ;Point to save area
294B Ø1ØAØØ               LD      BC,1Ø
294E EDBØ                 LDIR
295Ø C9                   RET
                   ;
2951 3E2C          PRMERR LD      A,44            ;Init "parameter error
2953 18E4                 JR      IOERR1
2955               PERR1  @@LOGOT                 ;Display and log
                          IFEQ    ØØH,1
                          LD      HL,
                          ENDIF
2955 3EØC                 LD      A,12
2957 EF                   RST     4Ø
2958 C34926               JP      $ABORT
                   ;
                   ;      Sector read routine
                   ;
295B D9            GETSEC EXX                     ;P/U alt registers
295C 7A                   LD      A,D             ;Any records left?
295D B3                   OR      E
295E 2ØØ5                 JR      NZ,NOTEND       ;Go if so
296Ø D9            BDEXT  EXX
2961 3E1C                 LD      A,1CH           ;EOF code
2963 A7                   AND     A               ;Set NZ condition
2964 C9                   RET
                   ;
2965 AF            NOTEND XOR     A               ;Check if used up ext
2966 BØ                   OR      B
2967 2Ø21                 JR      NZ,MORE         ;Go if not used up
2969 7E                   LD      A,(HL)          ;Check next trk#
296A FEFF                 CP      ØFFH            ;Non-allocated?
296C 28F2                 JR      Z,BDEXT         ;Then consider EOF
296E D5                   PUSH    DE              ;Save DE'
296F 56                   LD      D,(HL)          ;P/U track number
```

```
2970 23       Ø539Ø          INC     HL
2971 46       Ø540Ø          LD      B,(HL)          ;P/U other stuff
2972 23       Ø541Ø          INC     HL
2973 78       Ø542Ø          LD      A,B             ;Get starting gran
2974 Ø7       Ø543Ø          RLCA
2975 Ø7       Ø544Ø          RLCA                    ;Move to bits Ø-2
2976 Ø7       Ø545Ø          RLCA
2977 E6Ø7     Ø546Ø          AND     7               ;Mask off other garbage
2979 5F       Ø547Ø          LD      E,A             ;Multiply by 3
297A Ø7       Ø548Ø          RLCA
297B 83       Ø549Ø          ADD     A,E
297C 3C       Ø550Ø          INC     A               ;Offset from Ø
297D 5F       Ø551Ø          LD      E,A             ;  and move to E reg
297E ED534B2C Ø552Ø          LD      (TRKSEC),DE     ;Save for later
2982 D1       Ø553Ø          POP     DE              ;Restore DE'
2983 78       Ø554Ø          LD      A,B             ;Get number of grans
2984 E61F     Ø555Ø          AND     1FH
2986 47       Ø556Ø          LD      B,A             ;Multiply by 3
2987 Ø7       Ø557Ø          RLCA
2988 8Ø       Ø558Ø          ADD     A,B
2989 47       Ø559Ø          LD      B,A             ;And put in B reg
              Ø560Ø ;
              Ø561Ø ;        Read sector
              Ø562Ø ;
298A Ø5       Ø563Ø MORE     DEC     B               ;Count down # sec in ext
298B 1B       Ø564Ø          DEC     DE              ;Count down # records
298C D9       Ø565Ø          EXX                     ;Restore primary set
298D D5       Ø566Ø          PUSH    DE              ;Save DE
298E C5       Ø567Ø          PUSH    BC              ;Save BC
298F ED5B4B2C Ø568Ø          LD      DE,(TRKSEC)     ;P/U track and sector #
2993 3A492C   Ø569Ø          LD      A,(SDRIVE)      ;P/U source drive
2996 4F       Ø570Ø          LD      C,A
2997 FD36Ø712 Ø571Ø          LD      (IY+7),18       ;Reset sec/trk each time
299B          Ø572Ø          @@RDSEC                 ;Read sector to (HL)
299B 3E31     ØØØ76          LD      A,49
299D EF       ØØØ77          RST     4Ø
299E 28Ø5     Ø573Ø          JR      Z,OK2           ;Go if no errors
29AØ FEØ6     Ø574Ø          CP      6               ;  or address mark differs
29A2 C23629   Ø575Ø          JP      NZ,IOERR        ;Quit on any other
29A5 1C       Ø576Ø OK2      INC     E               ;Step to next sector
29A6 7B       Ø577Ø          LD      A,E
29A7 FE13     Ø578Ø          CP      19D             ;End of track?
29A9 2ØØ3     Ø579Ø          JR      NZ,NOTEOT       ;Go if not
29AB 1EØ1     Ø580Ø          LD      E,1             ;Reset to sector 1
29AD 14       Ø581Ø          INC     D               ;Next track
29AE ED534B2C Ø582Ø NOTEOT   LD      (TRKSEC),DE
29B2 C1       Ø583Ø          POP     BC
29B3 D1       Ø584Ø          POP     DE
29B4 AF       Ø585Ø          XOR     A
29B5 C9       Ø586Ø          RET
              Ø587Ø ;
              Ø588Ø ;        Parsing subroutines
              Ø589Ø ;
29B6 7E       Ø590Ø GETDRV2  LD      A,(HL)
29B7 FE3A     Ø591Ø          CP      ':'
29B9 3EFF     Ø592Ø          LD      A,ØFFH          ;'Not entered' value
29BB CØ       Ø593Ø          RET     NZ              ;If no second drive, give DIR
              Ø594Ø ;
29BC 7E       Ø595Ø GETDRV   LD      A,(HL)          ;Parse drivespec
29BD FE3A     Ø596Ø          CP      ':'
29BF 2Ø9Ø     Ø597Ø          JR      NZ,PRMERR       ;Go if missing
```

```
29C1 23         Ø598Ø          INC     HL
29C2 7E         Ø599Ø          LD      A,(HL)          ;P/U drivespec
29C3 FE3Ø       Ø6ØØØ          CP      'Ø'             ;Be sure digit
29C5 388A       Ø6Ø1Ø          JR      C,PRMERR
29C7 FE38       Ø6Ø2Ø          CP      '7'+1
29C9 3Ø86       Ø6Ø3Ø          JR      NC,PRMERR
29CB 23         Ø6Ø4Ø          INC     HL              ;Bump cmdline ptr
29CC E6Ø7       Ø6Ø5Ø          AND     7               ;Make drive # binary
29CE C9         Ø6Ø6Ø          RET
                Ø6Ø7Ø ;
29CF 7E         Ø6Ø8Ø SKIPSP   LD      A,(HL)          ;Skip spaces
29DØ FE2Ø       Ø6Ø9Ø          CP      ' '
29D2 CØ         Ø61ØØ          RET     NZ
29D3 23         Ø611Ø          INC     HL
29D4 18F9       Ø612Ø          JR      SKIPSP
                Ø613Ø ;
29D6 7E         Ø614Ø SKIPLT   LD      A,(HL)          ;Skip letters/digits/$
29D7 CDE829     Ø615Ø          CALL    CHKLET          ;Check letter/digit/$
29DA CØ         Ø616Ø          RET     NZ
29DB 23         Ø617Ø          INC     HL
29DC 18F8       Ø618Ø          JR      SKIPLT
                Ø619Ø ;
29DE 7E         Ø62ØØ MOVELT   LD      A,(HL)          ;Move letters/digits/$
29DF CDE829     Ø621Ø          CALL    CHKLET
29E2 CØ         Ø622Ø          RET     NZ
29E3 23         Ø623Ø          INC     HL              ;Inc from buffer
29E4 12         Ø624Ø          LD      (DE),A          ;Store
29E5 13         Ø625Ø          INC     DE              ;Inc to buffer
29E6 18F6       Ø626Ø          JR      MOVELT
                Ø627Ø ;
29E8 CB7F       Ø628Ø CHKLET   BIT     7,A             ;Graphic?
29EA CØ         Ø629Ø          RET     NZ
29EB FE61       Ø63ØØ          CP      'a'             ;Lowercase?
29ED 38Ø2       Ø631Ø          JR      C,NOTLC         ;Go if not
29EF CBAF       Ø632Ø          RES     5,A             ;  else make upper case
29F1 FE24       Ø633Ø NOTLC    CP      '$'             ;Dollar sign?
29F3 C8         Ø634Ø          RET     Z
29F4 FE3Ø       Ø635Ø          CP      'Ø'             ;Digit?
29F6 D8         Ø636Ø          RET     C               ;Return (NZ) if less
29F7 FE3A       Ø637Ø          CP      '9'+1
29F9 3ØØ2       Ø638Ø          JR      NC,NOTDIG       ;Go if not digit
29FB BF         Ø639Ø          CP      A               ;Mark as letter/digit/$
29FC C9         Ø64ØØ          RET
29FD FE41       Ø641Ø NOTDIG   CP      'A'             ;Letter?
29FF D8         Ø642Ø          RET     C               ;Return (NZ) if less
2AØØ FE5A       Ø643Ø          CP      'Z'
2AØ2 DØ         Ø644Ø          RET     NC              ;Z if =Z, NZ if >Z
2AØ3 BF         Ø645Ø          CP      A               ;Z if <Z
2AØ4 C9         Ø646Ø          RET
                Ø647Ø ;
2AØ5 E5         Ø648Ø CPHLDE   PUSH    HL              ;Compare HL and DE
2AØ6 A7         Ø649Ø          AND     A
2AØ7 ED52       Ø65ØØ          SBC     HL,DE
2AØ9 E1         Ø651Ø          POP     HL
2AØA C9         Ø652Ø          RET
                Ø653Ø ;
                Ø654Ø ;If NOT (-) spec given, reverse Z flag setting
                Ø655Ø ;
2AØB F5         Ø656Ø NOTCHK   PUSH    AF              ;Save current setting
2AØC 3AB72A     Ø657Ø          LD      A,(NOTPRM)      ;Was NOT entered?
2AØF B7         Ø658Ø          OR      A
```

```
2A1Ø 28Ø8     Ø659Ø         JR      Z,NOTNOT        ;No, restore previous
2A12 F1       Ø66ØØ         POP     AF              ;Get previous
2A13 28Ø2     Ø661Ø         JR      Z,SETIT         ;Was Z, make NZ
2A15 AF       Ø662Ø         XOR     A               ;  else was NZ, make Z
2A16 C9       Ø663Ø         RET
2A17 F6FF     Ø664Ø SETIT   OR      ØFFH            ;make NZ
2A19 C9       Ø665Ø         RET
2A1A F1       Ø666Ø NOTNOT  POP     AF              ;Get previous flags
2A1B C9       Ø667Ø         RET
              Ø668Ø ;
              Ø669Ø ;Display mod 3 TRSDOS disk directory
              Ø67ØØ ;
2A1C E5       Ø671Ø SHOW    PUSH    HL
2A1D D5       Ø672Ø         PUSH    DE
2A1E C5       Ø673Ø         PUSH    BC              ;Save registers
2A1F ØEØØ     Ø674Ø         LD      C,Ø             ;Init char count
2A21 21E92B   Ø675Ø         LD      HL,FNAME        ;=>name
2A24 7E       Ø676Ø NMDSP   LD      A,(HL)          ;Get a character
2A25 FEØ3     Ø677Ø         CP      ETX             ;Are we done?
2A27 28Ø7     Ø678Ø         JR      Z,NMEND         ;Finish if so
2A29 CD4E26   Ø679Ø         CALL    $DSP            ;Print this char
2A2C ØC       Ø68ØØ         INC     C               ;Count it
2A2D 23       Ø681Ø         INC     HL              ;=>next char
2A2E 18F4     Ø682Ø         JR      NMDSP           ;Until ETX
              Ø683Ø ;
2A3Ø 21ØØØØ   Ø684Ø NMEND   LD      HL,$-$          ;P/u line/char count
2A31          Ø685Ø CCOUNT  EQU     $-2
2A33 79       Ø686Ø         LD      A,C             ;Count for this entry
2A34 85       Ø687Ø         ADD     A,L             ;Add to previous
2A35 6F       Ø688Ø         LD      L,A             ;Save posn
2A36 3E1Ø     Ø689Ø         LD      A,16            ;Spaces for entry
2A38 91       Ø69ØØ         SUB     C               ;Less used
2A39 47       Ø691Ø         LD      B,A             ;Remaining to B
2A3A 3E2Ø     Ø692Ø SPLP    LD      A,' '           ;Pad remaining w/spaces
2A3C CD4E26   Ø693Ø         CALL    $DSP
2A3F 2C       Ø694Ø         INC     L               ;Count it
2A4Ø 7D       Ø695Ø         LD      A,L             ;Check char posn
2A41 FE4E     Ø696Ø         CP      78              ;End of line?
2A43 28Ø9     Ø697Ø         JR      Z,ELINE         ;Then print CR
2A45 1ØF3     Ø698Ø         DJNZ    SPLP            ;  else keep going
              Ø699Ø ;
2A47 22312A   Ø7ØØØ ESHOW   LD      (CCOUNT),HL     ;Save line/char posn
2A4A C1       Ø7Ø1Ø         POP     BC              ;Restore regs
2A4B D1       Ø7Ø2Ø         POP     DE
2A4C E1       Ø7Ø3Ø         POP     HL
2A4D C9       Ø7Ø4Ø         RET                     ;Done w/entry
              Ø7Ø5Ø ;
2A4E 3EØD     Ø7Ø6Ø ELINE   LD      A,CR            ;Hit end of line
2A5Ø CD4E26   Ø7Ø7Ø         CALL    $DSP
2A53 24       Ø7Ø8Ø         INC     H               ;Bump line posn
2A54 2EØØ     Ø7Ø9Ø         LD      L,Ø             ;Start on next
2A56 3E17     Ø71ØØ         LD      A,23            ;Max lines
2A58 BC       Ø711Ø         CP      H               ;There yet?
2A59 2ØEC     Ø712Ø         JR      NZ,ESHOW        ;Nope
2A5B          Ø713Ø         @@KEY                   ;Wait for a key
2A5B 3EØ1     ØØØ78         LD      A,1
2A5D EF       ØØØ79         RST     4Ø
2A5E CD662A   Ø714Ø         CALL    $CLS            ;Clear the display
2A61 21ØØØØ   Ø715Ø         LD      HL,Ø            ;Restart count
2A64 18E1     Ø716Ø         JR      ESHOW
              Ø717Ø ;
```

```
2A66 3E1C       Ø718Ø $CLS     LD      A,HOME          ;Cursor home
2A68 CD4E26     Ø719Ø          CALL    $DSP
2A6B 3E1F       Ø72ØØ          LD      A,CLR           ;Clear to end-of-frame
2A6D C34E26     Ø721Ø          JP      $DSP
                Ø722Ø ;
2A7Ø ØØ         Ø723Ø CKEARLY  DB      Ø
2A71 3A222D     Ø724Ø          LD      A,(DBUFF+22H)   ;Get type byte
2A74 FEFF       Ø725Ø          CP      ØFFH            ;Do we know this one?
2A76 C8         Ø726Ø          RET     Z               ;OK to continue
2A77 3A4A2C     Ø727Ø          LD      A,(DDRIVE)      ;Doesn't matter if
2A7A 3C         Ø728Ø          INC     A               ;  only doing DIR
2A7B C8         Ø729Ø          RET     Z
2A7C 218A2B     Ø73ØØ          LD      HL,EARLYD       ;Err msg
2A7F C35529     Ø731Ø          JP      PERR1           ;Quit
                Ø732Ø ;
2A82 8Ø         Ø733Ø PRMTBL$  DB      8ØH
2A83 55         Ø734Ø          DB      ABB!FLAG!5
2A84 51         Ø735Ø          DB      'QUERY',Ø
     55 45 52 59 ØØ
2A8A BF26       Ø736Ø          DW      QPARM+1
2A8C 53         Ø737Ø          DB      ABB!FLAG!3
2A8D 53         Ø738Ø          DB      'SYS',Ø
     59 53 ØØ
2A91 BØ26       Ø739Ø          DW      SPARM+1
2A93 53         Ø74ØØ          DB      ABB!FLAG!3
2A94 49         Ø741Ø          DB      'INV',Ø
     4E 56 ØØ
2A98 B626       Ø742Ø          DW      IPARM+1
2A9A 53         Ø743Ø          DB      ABB!FLAG!3
2A9B 56         Ø744Ø          DB      'VIS',Ø
     49 53 ØØ
2A9F B326       Ø745Ø          DW      VPARM+1
2AA1 53         Ø746Ø          DB      ABB!FLAG!3
2AA2 4F         Ø747Ø          DB      'OLD',Ø
     4C 44 ØØ
2AA6 C526       Ø748Ø          DW      OPARM+1
2AA8 53         Ø749Ø          DB      ABB!FLAG!3
2AA9 4E         Ø75ØØ          DB      'NEW',Ø
     45 57 ØØ
2AAD C226       Ø751Ø          DW      NPARM+1
2AAF 53         Ø752Ø          DB      ABB!FLAG!3
2ABØ 44         Ø753Ø          DB      'DIR',Ø
     49 52 ØØ
2AB4 A426       Ø754Ø          DW      DPARM+1
2AB6 ØØ         Ø755Ø          NOP
                Ø756Ø ;
                Ø757Ø ;        Messages and buffers
                Ø758Ø ;
2AB7 ØØ         Ø759Ø NOTPRM   DB      Ø
2AB8 24         Ø76ØØ PATTRN   DB      '$$$$$$$$'
     24 24 24 24 24 24 24
2ACØ 24         Ø761Ø PATEXT   DB      '$$$'
     24 24
2AC3 43         Ø762Ø HELLO$   DB      'CONV'
     4F 4E 56
2AC7            Ø763Ø *GET     CLIENT:3
                Ø395Ø ;CLIENTS/ASM - File to establish sign-on headers
                Ø396Ø ;
2AC7 2Ø         Ø397Ø          DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
2AF1 2Ø          Ø398Ø          DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                 Ø399Ø ;
2BØ6 41          Ø4ØØØ          DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
2B2E 2Ø          Ø4Ø1Ø          DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
2B46 53          Ø764Ø NOTONE   DB      'Source and Destination drives are the same',CR
     6F 75 72 63 65 2Ø 61 6E
     64 2Ø 44 65 73 74 69 6E
     61 74 69 6F 6E 2Ø 64 72
     69 76 65 73 2Ø 61 72 65
     2Ø 74 68 65 2Ø 73 61 6D
     65 ØD
2B71 53          Ø765Ø NOTØ     DB      'Source cannot be drive Ø',CR
     6F 75 72 63 65 2Ø 63 61
     6E 6E 6F 74 2Ø 62 65 2Ø
     64 72 69 76 65 2Ø 3Ø ØD
2B8A 43          Ø766Ø EARLYD   DB      'Cannot CONV Protected Diskette',CR
     61 6E 6E 6F 74 2Ø 43 4F
     4E 56 2Ø 5Ø 72 6F 74 65
     63 74 65 64 2Ø 44 69 73
     6B 65 74 74 65 ØD
2BA9 3F          Ø767Ø QMARK    DB      '? ',ETX
     2Ø Ø3
2BAC 2Ø          Ø768Ø EXISTQ   DB      '  File exists -- replace it? ',ETX
     2Ø 46 69 6C 65 2Ø 65 78
     69 73 74 73 2Ø 2D 2D 2Ø
     72 65 7Ø 6C 61 63 65 2Ø
     69 74 3F 2Ø Ø3
2BCA 43          Ø769Ø CONVS    DB      'Converting file: ',ETX
     6F 6E 76 65 72 74 69 6E
     67 2Ø 66 69 6C 65 3A 2Ø
     Ø3
2BDC 43          Ø77ØØ CONVQ    DB      'Convert file '
     6F 6E 76 65 72 74 2Ø 66
     69 6C 65 2Ø
ØØ2Ø             Ø771Ø FNAME    DS      32              ;Must follow CONVQ
ØØ2Ø             Ø772Ø FCB      DS      32              ;For INIT/WRITE
ØØ2Ø             Ø773Ø FCB2     DS      32              ;For OPEN (test for already existing)
ØØØ1             Ø774Ø SDRIVE   DS      1
ØØØ1             Ø775Ø DDRIVE   DS      1
ØØØ2             Ø776Ø TRKSEC   DS      2
ØØØ5             Ø777Ø ABUFF    DS      5
ØØØA             Ø778Ø SAVDCT   DS      1Ø
2DØØ             Ø779Ø          ORG     $<-8+1<8
1ØØØ             Ø78ØØ DBUFF    DS      1ØØØH           ;16 sectors of directory
3DØØ             Ø781Ø TBUFF    EQU     $               ;To end of memory
26ØØ             Ø782Ø          END     BEGIN
```

```
$ABORT          2649 $CLS           2A66 $DSP           264E
$EXIT           263F $QUIT          2642 @@1            ØØØØ
@@2             ØØØØ @@3            ØØØØ @@4            ØØØØ
@MOD2           ØØØØ @MOD4          FFFF ABB            ØØ1Ø
ABUFF           2C4D BDEXT          296Ø BEGIN          26ØØ
BEGINA          26Ø9 BYEBYE         2945 CCOUNT         2A31
CHECKQ          28ØF CHKLET         29E8 CKEARLY        2A7Ø
CKNEW           28Ø3 CLR            ØØ1F CONVQ          2BDC
CONVS           2BCA CPHLDE         2AØ5 CPLOOP         276A
CR              ØØØD DBUFF          2DØØ DDRIVE         2C4A
DPARM           26A3 DREAD          27ØE EARLY          28A8
EARLYD          2B8A ELINE          2A4E ELOOP          2723
ESHOW           2A47 ETX            ØØØ3 EXISTQ         2BAC
EXLOOP          279B FCB            2CØ9 FCB2           2C29
FLAG            ØØ4Ø FNAME          2BE9 GETDRV         29BC
GETDRV2         29B6 GETONE         28D8 GETSEC         295B
GOTEXT          27A5 GOTNAM         278C HELLO$         2AC3
HOME            ØØ1C INV            2759 IOERR          2936
IOERR1          2939 IPARM          26B5 KFLG           2724
LF              ØØØA MATCH          2773 MORE           298A
MOVELT          29DE MOVING         27C1 MVNAM1         266Ø
MVNAME          2782 MYHIGH         28D5 NMATCH         2776
NMDSP           2A24 NMEND          2A3Ø NOEXT          267F
NORO            27F4 NOSIV          276Ø NOTØ           2B71
NOTCHK          2AØB NOTCMDR        2622 NOTDIG         29FD
NOTEND          2965 NOTEOS         291C NOTEOT         29AE
NOTLC           29F1 NOTNOT         2A1A NOTONE         2B46
NOTPRM          2AB7 NOTSYS         274C NPARM          26C1
OKØ             26FD OK1            271C OK2            29A5
OK3             28C3 OPARM          26C4 PATEXT         2ACØ
PATTRN          2AB8 PERR1          2955 PGRM           2655
PRMERR          2951 PRMTBL$        2A82 QMARK          2BA9
QPARM           26BE QUERY          2828 QUIT           293Ø
READ            28CF SAVDCT         2C52 SDRIVE         2C49
SETIT           2A17 SFLG           27E4 SHOW           2A1C
SIV             273A SKIPIT         29ØF SKIPLT         29D6
SKIPSP          29CF SPARM          26AF SPLP           2A3A
STACK           2646 TAKEIT1        2869 TAKEIT2        2891
TBUFF           3DØØ TRKSEC         2C4B TRSDOS         289B
VPARM           26B2 WRDUN          28FF WRITE          28E8
WRLOOP          28EF @@ABORT        BØØE @@ADTSK        BØA1
@@BANK          B5B9 @@BKSP         B299 @@BREAK        B5CF
@@CHNIO         AFF9 @@CKBRKC       B61D @@CKDRV        BØF5
@@CKEOF         B2AE @@CKTSK        BØ8C @@CLOSE        B284
@@CLS           B6Ø7 @@CMNDI        BØ38 @@CMNDR        BØ4D
@@CTL           AE5D @@DATE         AFCF @@DCSTAT       B134
@@DEBUG         BØ77 @@DECHEX       B539 @@DIRRD        B4A6
@@DIRWR         B4BB @@DIV16        B524 @@DIV8         B5ØF
@@DODIR         B1ØA @@DSP          AE21 @@DSPLY        AEC1
@@ERROR         BØ62 @@EXIT         BØ23 @@FEXT         B413
@@FLAGS         B5A3 @@FNAME        B428 @@FSPEC        B3FE
@@GATRD         B491 @@GATWR        B4DØ @@GET          AE35
@@GTDCB         B452 @@GTDCT        B43D @@GTMOD        B467
@@HDFMT         B1DC @@HEX16        B578 @@HEX8         B563
@@HEXDEC        B54E @@HIGH$        B58D @@INIT         B25A
@@KBD           AE99 @@KEY          AEØD @@KEYIN        AEAD
@@KLTSK         BØEØ @@LOAD         B3D4 @@LOC          B2C3
@@LOF           B2D8 @@LOGER        AEF8 @@LOGOT        AFØD
@@MSG           AF44 @@MUL16        B4FA @@MUL8         B4E5
@@OPEN          B26F @@PARAM        AFBA @@PAUSE        AFA5
```

```
@@PEOF          B2ED @@POSN          B3Ø2 @@PRINT          AF59
@@PRT           AE71 @@PUT           AE49 @@RAMDIR         B11F
@@RDSEC         B1B2 @@RDSSC         B47C @@READ           B317
@@REMOV         B245 @@RENAM         B23Ø @@REW            B32C
@@RMTSK         BØB6 @@RPTSK         BØCB @@RREAD          B341
@@RSLCT         B19D @@RSTOR         B15E @@RUN            B3E9
@@RWRIT         B356 @@SEEK          B188 @@SEEKSC         B36B
@@SKIP          B38Ø @@SLCT          B149 @@STEPI          B173
@@TIME          AFE4 @@VDCTL         AF9Ø @@VER            B395
@@VRSEC         B1C7 @@WEOF          B3AA @@WHERE          AE85
@@WRITE         B3BF @@WRSEC         B1F1 @@WRSSC          B2Ø6
@@WRTRK         B21B
ØØØØØ Total errors
```

NOTES:

NOTES:

FLOPPY/DCT - 5 1/4" drive setup

The Floppy DCT program allows up to four physical 5 1/4" floppy drives to be assigned to the seven different logical drive positions. It is activated with the SYSTEM (DRIVER) Library command.

```
                ;LDOSDCT/ASM - Floppy Disk DCT
ØØØØ                    TITLE   <FLOPPY/DCT - LS-DOS 6.2>
                ;
                ;       Program installs a standard DCT into a logical
                ;       drive slot as specified by:
                ;        SYSTEM (DRIVE=d,DRIVER="LDOS")
                ;       The default DCT is taken from slot Ø of the
                ;       System Information Sector (7ØH-79H).
                ;
ØØØD            CR      EQU     13
ØØØA            LF      EQU     1Ø
                ;
ØØØØ            *GET    SVCMAC:3                ;SVC Macro equivalents
                ;SVCMAC/ASM - LS-DOS Version VI
                *LIST   OFF
                *LIST   ON
ØØØØ            *GET    COPYCOM:3               ;Copyright message
                ; COPYCOM - File for Copyright COMment block
                ;
ØØØØ                    COM     '<*(C) 1982,83,84 by LSI*>'
                ;
2CØØ                    ORG     2CØØH
                ;
                BEGIN
2CØØ                    @@CKBRKC
2CØØ 3E6A               LD      A,1Ø6
2CØ2 EF                 RST     4Ø
2CØ3 28Ø4               JR      Z,BEGINA        ;Continue if no break
2CØ5 21FFFF             LD      HL,-1           ; else abort
2CØ8 C9                 RET
                ;
2CØ9 D5         BEGINA  PUSH    DE              ;Save the DCT location
2CØA                    @@DSPLY HELLO$          ;Display the signon
                        IFEQ    Ø1H,1
2CØA 21E22C             LD      HL,HELLO$
                        ENDIF
2CØD 3EØA               LD      A,1Ø
2CØF EF                 RST     4Ø
2C1Ø D1                 POP     DE
2C11 7A         LDOS    LD      A,D             ;Make sure that a
2C12 B3                 OR      E               ;  drive # was entered
2C13 CAD52C             JP      Z,NODRV         ;Go if no drive
                ;
                ;       Check if entry from SET command
                ;
2C16                    @@FLAGS
2C16 3E65               LD      A,1Ø1
2C18 EF                 RST     4Ø
2C19 FDCBØ25E           BIT     3,(IY+'C'-'A')  ;System request?
2C1D CAC92C             JP      Z,VIASET        ;Exit if not
2C2Ø 1A                 LD      A,(DE)
2C21 FEC9               CP      ØC9H            ;Is drive disabled?
2C23 C2CD2C             JP      NZ,ACTIVE       ;Must be disabled
2C26 D5                 PUSH    DE              ;Save DCT address
2C27 CDB22C             CALL    GETCFG          ;Load sysinfo sector
2C2A C2BD2C             JP      NZ,IOERR        ;Quit on read error
2C2D FDCBØB66           BIT     4,(IY+'L'-'A')  ;Suppress 8" queries?
2C31 2Ø1C               JR      NZ,LDOS3        ;NZ=suppress
                ;
                ;       Query as to 5" or 8" floppy
                ;
```

```
2C33 21DB2D   DRVTYP  LD      HL,DRVTYP$      ;"Enter drive code...
2C36                  @@DSPLY
                      IFEQ    ØØH,1
                      LD      HL,
                      ENDIF
2C36 3EØA             LD      A,1Ø
2C38 EF               RST     4Ø
2C39 21312E           LD      HL,BUF          ;Pt to buffer
2C3C Ø1ØØØ1           LD      BC,1<8          ;Allow 1 char only
2C3F                  @@KEYIN                 ;Get response
2C3F 3EØ9             LD      A,9
2C41 EF               RST     4Ø
2C42 DAD12C           JP      C,BREAK         ;Quit on Break
2C45 7E               LD      A,(HL)          ;P/u char response
2C46 D63Ø             SUB     'Ø'             ;Adjust to binary
2C48 FEØ2             CP      2               ;Make sure requested
2C4A 3ØE7             JR      NC,DRVTYP       ;  type is supported
2C4C 32722C           LD      (LX8Ø5+1),A
              ;
              ;       Prompt user for physical drive address
              ;
2C4F          LDOS3   @@DSPLY DRVADR$         ;"Enter physical...
                      IFEQ    Ø1H,1
2C4F 21FF2D           LD      HL,DRVADR$
                      ENDIF
2C52 3EØA             LD      A,1Ø
2C54 EF               RST     4Ø
2C55 21312E           LD      HL,BUF          ;Input buffer
2C58 Ø1ØØØ1           LD      BC,1<8          ;Allow 1 char only
2C5B                  @@KEYIN                 ;Get response
2C5B 3EØ9             LD      A,9
2C5D EF               RST     4Ø
2C5E DAD12C           JP      C,BREAK         ;Quit on Break
2C61 7E               LD      A,(HL)          ;P/u the response
2C62 D631             SUB     '1'             ;Adjust to binary
2C64 FEØ4             CP      3+1             ;Be sure in range
2C66 3ØE7             JR      NC,LDOS3        ;Redo if not
              ;
              ;       Convert drive address to select code
              ;
2C68 FEØ3             CP      3               ;Convert 3 to 4
2C6A 3F               CCF
2C6B CEØØ             ADC     A,Ø
2C6D FEØ1             CP      1               ;Convert <Ø,1,2,4>
2C6F 17               RLA                     ;  to <1, 2, 4, 8>
2C7Ø 47               LD      B,A             ;Hang on to request
              ;
              ;       Index the default drive code table
              ;
2C71          LX8Ø5   EQU     $
                      IF      @MOD2
                      LD      A,1             ;8"
                      ENDIF
                      IF      @MOD4
2C71 3EØØ             LD      A,Ø             ;5"
                      ENDIF
2C73 4F               LD      C,A
2C74 87               ADD     A,A             ;Times 2
2C75 81               ADD     A,C             ;Times 3
2C76 87               ADD     A,A             ;Times 6
2C77 81               ADD     A,C             ;Times 7
```

```
2C78 21232E                LD      HL,DRVTAB$      ;Index into 5" or 8"
2C7B 85                    ADD     A,L             ;  default table
2C7C 6F                    LD      L,A
2C7D 8C                    ADC     A,H
2C7E 95                    SUB     L
2C7F 67                    LD      H,A
2C8Ø 23                    INC     HL
2C81 7E                    LD      A,(HL)          ;P/u default DCT+4
2C82 E6FØ                  AND     ØFØH            ;Remove drive select
2C84 BØ                    OR      B               ;Merge in new one
2C85 77                    LD      (HL),A          ;Update the DCT
2C86 2B                    DEC     HL
2C87 Ø1Ø7ØØ                LD      BC,7            ;Init for 7-byte move
2C8A D1                    POP     DE              ;DE => DCT$
2C8B D5                    PUSH    DE              ;Save DCT$ pointer
2C8C 13                    INC     DE
2C8D 13                    INC     DE
2C8E 13                    INC     DE              ;Index to DCT+3
2C8F EDBØ                  LDIR
2C91 D1                    POP     DE
2C92 D5                    PUSH    DE              ;Save start again
2C93 217Ø2F                LD      HL,BUFFER+7ØH   ;Index the default vector
2C96 ØEØ3                  LD      C,3             ;Move in driver vector
2C98 EDBØ                  LDIR
2C9A D1                    POP     DE
                    ;
                    ;      Compute the actual drive number used
                    ;
2C9B                       @@GTDCT                 ;Get drive Ø(ldir set C=Ø
2C9B 3E51                  LD      A,81
2C9D EF                    RST     4Ø
2C9E FDE5                  PUSH    IY              ;Pass to HL for sub
2CAØ E1                    POP     HL              ;HL => start DCT's
2CA1 EB                    EX      DE,HL           ;DE=start, HL=current
2CA2 B7                    OR      A               ;Clear carry
2CA3 ED52                  SBC     HL,DE           ;HL = offset from start
2CA5 ØEØA                  LD      C,1Ø            ;DCT length
2CA7                       @@DIV16                 ;HL+A = HL/C
2CA7 3E5E                  LD      A,94
2CA9 EF                    RST     4Ø
2CAA 4D                    LD      C,L             ;Result = drive #
2CAB                       @@RSTOR                 ;Restore drive
2CAB 3E2C                  LD      A,44
2CAD EF                    RST     4Ø
2CAE 21ØØØØ                LD      HL,Ø            ;Set no error return
2CB1 C9                    RET                     ;Init complete
                    ;
                    ;      Routines to read/write the config sector
                    ;
2CB2 21ØØ2F         GETCFG LD      HL,BUFFER       ;Use buffer for I/O
                    ;
                           IF      @MOD2
                           LD      C,L             ;Pass drive #
                           PUSH    IY              ;Save IY
2CB5                       @@GTDCT                 ;Fetch DCT
                           LD      A,81
                           RST     4Ø
                           LD      A,(IY+3)        ;Get data
                           AND     28H             ;Bit 5/3
                           CP      2ØH             ;8" floppy?
                           JR      NZ,SETSYS1      ;Go if not
```

```
                        LD      A,(IY+4)          ;Fetch data
                        AND     5ØH               ;Bit 6/4
                        CP      4ØH               ;DD not alien?
                        JR      NZ,SETSYS1        ;Go if not
                        LD      D,(IY+9)          ;Get dir cyl
                        LD      E,L               ;Sector Ø
2CB5                    @@RDSEC                   ;Read sector
                        LD      A,49
                        RST     4Ø
                        CP      6                 ;Directory?
                        JR      NZ,SETSYS2        ;Nope, error
                        LD      A,(BUFFER+ØCDH)   ;Get GAT data
                        BIT     7,A               ;System disk?
                SETSYS1 LD      DE,Ø<8+2          ;Normal sysinfo sector
                        JR      NZ,$+3            ;Go if data disk
                        INC     D                 ;  else sysinfo on 1
                        XOR     A                 ;Set Z for no error
                SETSYS2 POP     IY                ;Restore DCT
                        RET     NZ                ;Go if error
                        ENDIF
                ;
                        IF      @MOD4
2CB5 11Ø2ØØ             LD      DE,Ø<8+2          ;Get Config sector
                        ENDIF
2CB8 4D                 LD      C,L               ;  of system drive
2CB9                    @@RDSEC                   ;Read it into core
2CB9 3E31               LD      A,49
2CBB EF                 RST     4Ø
2CBC C9                 RET
                ;
2CBD 6F         IOERR   LD      L,A               ;Error # to HL
2CBE 26ØØ               LD      H,Ø
2CCØ F6CØ               OR      ØCØH              ;Abbrev, return
2CC2                    @@ERROR                   ;Display the error
2CC2 3E1A               LD      A,26
2CC4 EF                 RST     4Ø
2CC5                    @@CKBRKC                  ;Clear any Break
2CC5 3E6A               LD      A,1Ø6
2CC7 EF                 RST     4Ø
2CC8 C9                 RET
                ;
                ;       Internal error display routine
                ;
2CC9 216E2D     VIASET  LD      HL,VIASET$        ;"Install with SYSTEM
2CCC DD                 DB      ØDDH
2CCD 218F2D     ACTIVE  LD      HL,ACTIVE$        ;"Drive in use
2CDØ DD                 DB      ØDDH
2CD1 21CB2D     BREAK   LD      HL,BREAK$         ;"Command aborted
2CD4 DD                 DB      ØDDH
2CD5 21AD2D     NODRV   LD      HL,NODRV$         ;"Need a drive #
2CD8                    @@LOGOT
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
2CD8 3EØC               LD      A,12
2CDA EF                 RST     4Ø
2CDB 21FFFF             LD      HL,-1             ;Set abort code
2CDE                    @@CKBRKC                  ;Clear any break
2CDE 3E6A               LD      A,1Ø6
2CEØ EF                 RST     4Ø
2CE1 C9                 RET
```

```
               Ø2Ø2Ø ;
2CE2 ØA        Ø2Ø3Ø HELLO$  DB      LF,'FLOPPY Setup'
     46 4C 4F 5Ø 5Ø 59 2Ø 53
     65 74 75 7Ø
2CEF           Ø2Ø4Ø *GET    CLIENT:3
               Ø395Ø ;CLIENTS/ASM - File to establish sign-on headers
               Ø396Ø ;
2CEF 2Ø        Ø397Ø         DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
2D19 2Ø        Ø398Ø         DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
               Ø399Ø ;
2D2E 41        Ø4ØØØ         DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
2D56 2Ø        Ø4Ø1Ø         DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
               Ø2Ø5Ø ;
2D6E 4D        Ø2Ø6Ø VIASET$ DB      'Must install via SYSTEM (DRIVER=',CR
     75 73 74 2Ø 69 6E 73 74
     61 6C 6C 2Ø 76 69 61 2Ø
     53 59 53 54 45 4D 2Ø 28
     44 52 49 56 45 52 3D ØD
2D8F 44        Ø2Ø7Ø ACTIVE$ DB      'Drive slot is already enabled',CR
     72 69 76 65 2Ø 73 6C 6F
     74 2Ø 69 73 2Ø 61 6C 72
     65 61 64 79 2Ø 65 6E 61
     62 6C 65 64 ØD
2DAD 4C        Ø2Ø8Ø NODRV$  DB      'Logical drive number required',CR
     6F 67 69 63 61 6C 2Ø 64
     72 69 76 65 2Ø 6E 75 6D
     62 65 72 2Ø 72 65 71 75
     69 72 65 64 ØD
2DCB 43        Ø2Ø9Ø BREAK$  DB      'Command aborted',CR
     6F 6D 6D 61 6E 64 2Ø 61
     62 6F 72 74 65 64 ØD
2DDB 2Ø        Ø21ØØ DRVTYP$ DB      '   Enter drive code (Ø=5", 1=8") > ',3
     2Ø 2Ø 45 6E 74 65 72 2Ø
     64 72 69 76 65 2Ø 63 6F
     64 65 2Ø 28 3Ø 3D 35 22
     2C 2Ø 31 3D 38 22 29 2Ø
     3E 2Ø Ø3
2DFF 2Ø        Ø211Ø DRVADR$ DB      '   Enter drive I/O address <1-4> > ',3
     2Ø 2Ø 45 6E 74 65 72 2Ø
     64 72 69 76 65 2Ø 49 2F
     4F 2Ø 61 64 64 72 65 73
     73 2Ø 3C 31 2D 34 3E 2Ø
     3E 2Ø Ø3
               Ø212Ø DRVTAB$
```

```
                 ;
                 ;        5" drive table
                 ;
2E23 44                   DB      Ø1ØØØ1ØØB        ;5", 6ms, delay=n
2E24 4Ø                   DB      Ø1ØØØØØØB        ;DDEN
2E25 FF                   DB      ØFFH             ;Start cylinder
2E26 27                   DB      4Ø-1             ;4Ø track drive
2E27 11                   DB      18-1             ;18 sec per cyl
2E28 45                   DB      3-1<5+6-1        ;6 sec/gran, 3 gran/cyl
2E29 14                   DB      4Ø/2             ;Directory track
                 ;
                 ;        8" table
                 ;
                          IF      @MOD4
2E2A 21                   DB      ØØ1ØØØØ1B        ;8", 6ms step
2E2B 4Ø                   DB      Ø1ØØØØØØB        ;DDEN
2E2C FF                   DB      ØFFH             ;Start cylinder
2E2D 4C                   DB      77-1             ;77 track drive
2E2E ØF                   DB      16-1             ;16 sec per cyl
2E2F 27                   DB      2-1<5+8-1        ;8 sec/gran, 2 gran/cyl
2E3Ø 26                   DB      77/2             ;Directory track
                          ENDIF
                 ;
                          IF      @MOD2
                          DB      Ø11ØØØ1ØB        ;+3 - 8", DD, 1Øms, delay
                          DB      Ø1ØØØØØØB        ;+4 - DDen capable
                          DB      4CH              ;+5 - current cyl
                          DB      77-1             ;+6 - high cylinder
                          DB      Ø<5+29           ;+7 - sides + high sec
                          DB      2<5+9            ;+8 - grans/cyl + sec/grn
                          DB      77/2             ;+9 - dir cylinder
                          ENDIF
                 ;
ØØØ2             BUF      DS      2
2FØØ                      ORG     $<-8+1<+8
Ø1ØØ             BUFFER   DS      256
                 ;
2CØØ                      END     BEGIN
```

```
@@1                 ØØØØ @@2                 ØØØØ @@3                 ØØØØ
@@4                 ØØØØ @MOD2               ØØØØ @MOD4               FFFF
ACTIVE              2CCD ACTIVE$             2D8F BEGIN               2CØØ
BEGINA              2CØ9 BREAK               2CD1 BREAK$              2DCB
BUF                 2E31 BUFFER              2FØØ CR                  ØØØD
DRVADR$             2DFF DRVTAB$             2E23 DRVTYP              2C33
DRVTYP$             2DDB GETCFG              2CB2 HELLO$              2CE2
IOERR               2CBD LDOS                2C11 LDOS3               2C4F
LF                  ØØØA LX8Ø5               2C71 NODRV               2CD5
NODRV$              2DAD VIASET              2CC9 VIASET$             2D6E
@@ABORT             81D9 @@ADTSK             826C @@BANK              8784
@@BKSP              8464 @@BREAK             879A @@CHNIO             81C4
@@CKBRKC            87E8 @@CKDRV             82CØ @@CKEOF             8479
@@CKTSK             8257 @@CLOSE             844F @@CLS               87D2
@@CMNDI             82Ø3 @@CMNDR             8218 @@CTL               8Ø28
@@DATE              819A @@DCSTAT            82FF @@DEBUG             8242
@@DECHEX            87Ø4 @@DIRRD             8671 @@DIRWR             8686
@@DIV16             86EF @@DIV8              86DA @@DODIR             82D5
@@DSP               7FEC @@DSPLY             8Ø8C @@ERROR             822D
@@EXIT              81EE @@FEXT              85DE @@FLAGS             876E
@@FNAME             85F3 @@FSPEC             85C9 @@GATRD             865C
@@GATWR             869B @@GET               8ØØØ @@GTDCB             861D
@@GTDCT             86Ø8 @@GTMOD             8632 @@HDFMT             83A7
@@HEX16             8743 @@HEX8              872E @@HEXDEC            8719
@@HIGH$             8758 @@INIT              8425 @@KBD               8Ø64
@@KEY               7FD8 @@KEYIN             8Ø78 @@KLTSK             82AB
@@LOAD              859F @@LOC               848E @@LOF               84A3
@@LOGER             8ØC3 @@LOGOT             8ØD8 @@MSG               81ØF
@@MUL16             86C5 @@MUL8              86BØ @@OPEN              843A
@@PARAM             8185 @@PAUSE             817Ø @@PEOF              84B8
@@POSN              84CD @@PRINT             8124 @@PRT               8Ø3C
@@PUT               8Ø14 @@RAMDIR            82EA @@RDSEC             837D
@@RDSSC             8647 @@READ              84E2 @@REMOV             841Ø
@@RENAM             83FB @@REW               84F7 @@RMTSK             8281
@@RPTSK             8296 @@RREAD             85ØC @@RSLCT             8368
@@RSTOR             8329 @@RUN               85B4 @@RWRIT             8521
@@SEEK              8353 @@SEEKSC            8536 @@SKIP              854B
@@SLCT              8314 @@STEPI             833E @@TIME              81AF
@@VDCTL             815B @@VER               856Ø @@VRSEC             8392
@@WEOF              8575 @@WHERE             8Ø5Ø @@WRITE             858A
@@WRSEC             83BC @@WRSSC             83D1 @@WRTRK             83E6

2CØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

FORMAT/CMD - Disk initialization program

The Format utility allows a floppy or hard disk to be initialized. Parameters are available to set the number of sides and cylinders, select the density, and set a boot step rate for system disks.

```
                ;FORMAT1/ASM - Format Program
0000                     TITLE   <FORMAT - LS-DOS 6.2>
0000                     SUBTTL  '<Format Execution Code>'
                ;
42E0            PASSWORD         EQU     42E0H
0062            RLS      EQU     62H
000A            LF       EQU     10
000D            CR       EQU     13
3C00            CRT3     EQU     3C00H
F800            CRT4     EQU     0F800H
                ;
0000            *GET     SVCMAC:3                ;SVC Macro equivalents
                ;SVCMAC/ASM - LS-DOS Version VI
                *LIST    OFF
                *LIST    ON
0000            *GET     COPYCOM:3               ;Copyright message
                ; COPYCOM - File for Copyright COMment block
                ;
0000                     COM     '<*(C) 1982,83,84 by LSI*>'
                *LIST    ON
                ;
2600                     ORG     2600H
                ;
                         IF      @MOD4
2600 9D         BOOTST$  DB      9DH             ;Boot step rate offset
                         ENDIF
                         IF      @MOD2
                BOOTST$  DB      03H
                         ENDIF
                ;
2601 FDCB0466   GOFMT    BIT     4,(IY+4)        ;Jump if alien controller
2605 C2C127              JP      NZ,HRDRV
2608 110000     FMTTBL   LD      DE,0            ;P/u table pointer
260B 1A                  LD      A,(DE)          ;P/u # of sectors to fmt
260C 13                  INC     DE              ;Adj for zero offset
260D 32E92A              LD      (SECTRK),A
2610 47                  LD      B,A
2611 FDCB046E            BIT     5,(IY+4)        ;Need twice as many
2615 2801                JR      Z,$+3           ;  if 2-sided drive
2617 07                  RLCA
2618 32E82A              LD      (SECCYL),A
261B 210000     SYSPRM   LD      HL,0            ;P/u system info parm
261E 7C                  LD      A,H             ;Don't format if system
261F B5                  OR      L               ;  info only is req
2620 C29427              JP      NZ,MOVFREE
2623 1A                  LD      A,(DE)          ;P/u track skew
2624 13                  INC     DE
2625 320327              LD      (TRKSKEW+1),A
2628 ED53A226            LD      (SECSKEW+1),DE  ;Format sector skew
                ;
                ;        Index past sector info
                ;
262C 3C                  INC     A               ;Add DE -> begin of sec #
262D 80                  ADD     A,B             ;B -> # of sectors/side
262E 83                  ADD     A,E             ; A+1 -> a code byte
262F 5F                  LD      E,A
2630 8A                  ADC     A,D
2631 93                  SUB     E
2632 57                  LD      D,A
2633 210031              LD      HL,FORMAT       ;Buffer for format data
2636 010030              LD      BC,HITBUF       ;Tempy ptrs to trk,sect info
```

Format Execution Code

```
                ;
                ;       Create the formatting data without trk,sect info
                ;
2639 1A         FMTDAT  LD      A,(DE)          ;P/u table format byte
263A 13                 INC     DE              ;Bump table ptr
263B FEF1               CP      ØF1H            ;Start of cylinder?
263D 282A               JR      Z,CODF1
263F FEF2               CP      ØF2H            ;Start of track trailer?
2641 282D               JR      Z,CODF2
2643 FEF3               CP      ØF3H            ;Start of track ID info?
2645 2833               JR      Z,CODF3
2647 FEF4               CP      ØF4H            ;End of table parms?
2649 2837               JR      Z,CODF4
264B FEF5               CP      ØF5H            ;Start of data?
264D C5                 PUSH    BC
264E 2ØØF               JR      NZ,CODE1        ;Go if not
                ;
                ;       Write 2 byte data pattern to format buffer
                ;
265Ø 1A                 LD      A,(DE)          ;P/u length to write
2651 13                 INC     DE              ;Bump to 1st data byte
2652 47                 LD      B,A             ;Xfer length to B
2653 1A                 LD      A,(DE)          ;P/u a data byte
2654 13                 INC     DE              ;Bump again for 2nd byte
2655 4F                 LD      C,A             ;Xfer 1st byte
2656 1A                 LD      A,(DE)          ;P/u 2nd byte
2657 71         CODF5   LD      (HL),C          ;Stuff into buf
2658 23                 INC     HL
2659 77                 LD      (HL),A
265A 23                 INC     HL
265B 1ØFA               DJNZ    CODF5           ;Loop til xfered
265D 18Ø6               JR      CODRET
                ;
                ;       Xfer bytes to the format buffer area
                ;       A => count to move
                ;       DE=> data byte to duplicate
                ;
265F 47         CODE1   LD      B,A             ;Count to B
266Ø 1A                 LD      A,(DE)          ;P/u data byte to move
2661 77         CODE1A  LD      (HL),A          ;Fill buf with byte
2662 23                 INC     HL
2663 1ØFC               DJNZ    CODE1A          ;Loop til done
2665 C1         CODRET  POP     BC
2666 13                 INC     DE              ;Bump table ptr
2667 18DØ               JR      FMTDAT          ;Back for more
                ;
                ;       Save the current table posn and the number of
                ;       sectors per cylinder on the stack.
                ;
2669 3AE92A     CODF1   LD      A,(SECTRK)      ;P/u # of sectors/side
266C D5         CODF1A  PUSH    DE              ;Save table pointer
266D F5                 PUSH    AF              ;Save value
266E 18C9               JR      FMTDAT
                ;
                ;       Done with a sector. Are there more on this cyl?
                ;
267Ø F1         CODF2   POP     AF              ;Count down the # of
2671 3D                 DEC     A               ;  sectors to format
2672 28Ø3               JR      Z,CODF2A        ;Go if last one done
```

Format Execution Code

```
2674 D1                 POP     DE              ;Recover table ptr
2675 18F5               JR      CODF1A          ;Loop for more
              ;
2677 F1       CODF2A    POP     AF              ;Clean the stack
2678 18BF               JR      FMTDAT          ;  and finish off the cyl
              ;
              ;         Build a table of the location in the format buffer of
              ;         the track and sector ID bytes, to be filled in during
              ;         the actual formatting.
              ;
267A 7D       CODF3     LD      A,L             ;Stuff pointer to where
267B Ø2                 LD      (BC),A          ;  track & sector info
267C Ø3                 INC     BC              ;  is to be placed
267D 7C                 LD      A,H
267E Ø2                 LD      (BC),A
267F Ø3                 INC     BC
268Ø 18B7               JR      FMTDAT
              ;
              ;         Finished building format cyl info. Terminate the ID table
              ;         with an extra 256 bytes in case of overrun.
              ;
2682 ED536527 CODF4     LD      (VERSKEW+1),DE  ;Table posn of verify order
2686 AF                 XOR     A               ;Stuff two X'ØØ's to
2687 Ø2                 LD      (BC),A          ;  indicate the end
2688 Ø3                 INC     BC              ;  of the ID posn table
2689 Ø2                 LD      (BC),A
268A Ø6ØØ               LD      B,Ø             ;Stuff 256 FF's into the
268C 3EFF               LD      A,ØFFH          ;  format buffer
268E 77                 LD      (HL),A
268F 23                 INC     HL
269Ø 1ØFC               DJNZ    $-2
              ;
              ;         Begin the formatting
              ;
2692                    @@DSPLY FMTCYL$         ;"formatting clinder...
                        IFEQ    Ø1H,1
2692 21392C             LD      HL,FMTCYL$
                        ENDIF
2695 3EØA               LD      A,1Ø
2697 EF                 RST     4Ø
2698 FD7EØ5   BGNFMT    LD      A,(IY+5)        ;P/u cylinder position
269B CD5Ø2A             CALL    CVDEC           ;Cvrt to decimal
269E CD892A             CALL    DSPCYL
26A1 Ø1ØØØØ   SECSKEW   LD      BC,Ø            ;Begin of sector table
26A4 21ØØ3Ø   BFMT1     LD      HL,HITBUF       ;P/u ptr to ID posn table
              ;
              BFMT2
26A7                    @@CKBRKC                ;Check for break
26A7 3E6A               LD      A,1Ø6
26A9 EF                 RST     4Ø
26AA C2B929             JP      NZ,BREAK        ;Go if so
              ;
26AD 5E                 LD      E,(HL)          ;P/u positions having
26AE 23                 INC     HL              ;  sector & cylinder
26AF 56                 LD      D,(HL)          ;  info to be stuffed
26BØ 23                 INC     HL              ;  into format data
26B1 7A                 LD      A,D             ;Finished?
26B2 B3                 OR      E
26B3 282Ø               JR      Z,BFMT4
```

Format Execution Code

```
26B5 FD7EØ5    Ø176Ø           LD      A,(IY+5)         ;P/u cylinder # & stuff
26B8 12        Ø177Ø           LD      (DE),A           ;  into format data
26B9 13        Ø178Ø           INC     DE
26BA FD7EØ3    Ø179Ø           LD      A,(IY+3)         ;Stuff the side-select
26BD E61Ø      Ø18ØØ           AND     1ØH              ;  bit
26BF ØF        Ø181Ø           RRCA
26CØ ØF        Ø182Ø           RRCA
26C1 ØF        Ø183Ø           RRCA
26C2 ØF        Ø184Ø           RRCA
26C3 12        Ø185Ø           LD      (DE),A           ;  into the format data
26C4 13        Ø186Ø           INC     DE
26C5 ØA        Ø187Ø           LD      A,(BC)           ;P/u the sector number
26C6 B7        Ø188Ø           OR      A
26C7 F2CF26    Ø189Ø           JP      P,BFMT3          ;Go if a good number
26CA 81        Ø19ØØ           ADD     A,C              ;  else off the end,
26CB 4F        Ø191Ø           LD      C,A              ;  calculate the beginning
26CC 38Ø1      Ø192Ø           JR      C,BFMT3          ;  of the sector table
26CE Ø5        Ø193Ø           DEC     B
26CF ØA        Ø194Ø BFMT3     LD      A,(BC)           ;P/u the next sector #
26DØ 12        Ø195Ø           LD      (DE),A           ;  and stuff in format data
26D1 13        Ø196Ø           INC     DE
26D2 Ø3        Ø197Ø           INC     BC
26D3 18D2      Ø198Ø           JR      BFMT2            ;Loop until cylinder done
               Ø199Ø ;
26D5 ED43A226  Ø2ØØØ BFMT4     LD      (SECSKEW+1),BC   ;Save end of sector table
26D9 FD56Ø5    Ø2Ø1Ø           LD      D,(IY+5)         ;P/u current cylinder
26DC 21ØØ31    Ø2Ø2Ø           LD      HL,FORMAT        ;Pt to format data
26DF CDFA29    Ø2Ø3Ø           CALL    SELECT           ;Drive select
26E2 C2A529    Ø2Ø4Ø           JP      NZ,IOERR         ;Go on error
26E5 CDØE2A    Ø2Ø5Ø           CALL    WRCYL            ;Cylinder write
26E8 C2A529    Ø2Ø6Ø           JP      NZ,IOERR
26EB FDCBØ46E  Ø2Ø7Ø           BIT     5,(IY+4)         ;Double sided?
26EF 28ØD      Ø2Ø8Ø           JR      Z,BFMT5
26F1 FDCBØ366  Ø2Ø9Ø           BIT     4,(IY+3)         ;Flip bit for 2nd side
26F5 2ØØ7      Ø21ØØ           JR      NZ,BFMT5         ;  if not already on it,
26F7 FDCBØ3E6  Ø211Ø           SET     4,(IY+3)         ;  else go to next
26FB Ø3        Ø212Ø           INC     BC               ;Bump to start side 2
26FC 18A6      Ø213Ø           JR      BFMT1            ;  at different sector #
26FE FDCBØ3A6  Ø214Ø BFMT5     RES     4,(IY+3)         ;Turn off side 2
27Ø2 3EØØ      Ø215Ø TRKSKEW   LD      A,Ø              ;P/u the track skew byte
27Ø4 81        Ø216Ø           ADD     A,C              ;Repoint to beginning
27Ø5 4F        Ø217Ø           LD      C,A              ;  of sector table
27Ø6 88        Ø218Ø           ADC     A,B              ;Skew start of next track
27Ø7 91        Ø219Ø           SUB     C
27Ø8 47        Ø22ØØ           LD      B,A
27Ø9 ED43A226  Ø221Ø           LD      (SECSKEW+1),BC
27ØD FD7EØ5    Ø222Ø           LD      A,(IY+5)         ;Finished?
271Ø FDBEØ6    Ø223Ø           CP      (IY+6)
2713 282Ø      Ø224Ø           JR      Z,BGNVER         ;Begin verify if so
2715 Ø142ØØ    Ø225Ø           LD      BC,1ØØØ/15       ;Approx 1 ms pause
2718           Ø226Ø           @@PAUSE                  ;  before STEPIN
2718 3E1Ø      ØØØØ8           LD      A,16
271A EF        ØØØØ9           RST     4Ø
271B CDØ42A    Ø227Ø           CALL    STEPIN           ;Step in
271E C2A529    Ø228Ø           JP      NZ,IOERR         ;Go on error
2721 Ø19826    Ø229Ø           LD      BC,BGNFMT        ;Place RET addr on stack
2724 CDØ92A    Ø23ØØ CKWAIT    CALL    RSELCT           ;Wait for idle FDC
2727 C2A529    Ø231Ø           JP      NZ,IOERR         ;Go on error
272A C5        Ø232Ø           PUSH    BC               ;Save RET addr
```

Format Execution Code

```
                02330 ;
                02340 ;       WAIT parameter for time delay after STEPIN
                02350 ;
272B 01C800     02360 WAITPRM LD      BC,3000/15       ;Approx 3 ms delay
272E 78         02370         LD      A,B              ;  after STEPIN
272F B1         02380         OR      C
2730 C8         02390         RET     Z                ;Do next track if no wait
2731            02400         @@PAUSE                  ;  else wait for count
2731 3E10       00010         LD      A,16
2733 EF         00011         RST     40
2734 C9         02410         RET
                02420 ;
                02430 ;       Begin the verification process
                02440 ;
2735 0E0D       02450 BGNVER  LD      C,CR             ;Posn to next dsply line
2737            02460         @@DSP
2737 3E02       00012         LD      A,2
2739 EF         00013         RST     40
273A CDFF29     02470         CALL    RESTOR           ;Restore to cyl 0
273D 206A       02480         JR      NZ,BVER9         ;Go on error
273F            02490         @@DSPLY VERCYL$          ;"verifying cylinder...
                00014         IFEQ    01H,1
273F 21512C     00015         LD      HL,VERCYL$
                00016         ENDIF
2742 3E0A       00017         LD      A,10
2744 EF         00018         RST     40
2745 1600       02500         LD      D,0              ;Init track count
                02510 BVER1
2747            02520         @@CKBRKC                 ;Check for break
2747 3E6A       00019         LD      A,106
2749 EF         00020         RST     40
274A C2B929     02530         JP      NZ,BREAK         ; and abort if so
                02540 ;
274D 6A         02550         LD      L,D              ;Pt to GAT byte for this
274E 262E       02560         LD      H,GATBUF<-8      ;  track & bypass verify
2750 7E         02570         LD      A,(HL)           ;  if track not formatted
2751 3C         02580         INC     A
2752 2836       02590         JR      Z,BVER8
                02600 ;
2754 7A         02610         LD      A,D
2755 CD502A     02620         CALL    CVDEC            ;Convert cyl # to ASCII
2758 D5         02630         PUSH    DE
2759 CD892A     02640         CALL    DSPCYL           ;Display the current cyl
275C D1         02650         POP     DE
275D AF         02660         XOR     A                ;Initialize starting sector
275E 327227     02670         LD      (BVER5+1),A
2761 326927     02680         LD      (BVER4+1),A
2764 010000     02690 VERSKEW LD      BC,0             ;P/u start of sector tbl
2767 0A         02700 BVER3   LD      A,(BC)           ;P/u sector #
2768 C600       02710 BVER4   ADD     A,0              ;Add in a side's sectors
276A 5F         02720         LD      E,A              ;  if on side 2
276B CD272A     02730         CALL    VERSEC           ;Sector verify
276E 2039       02740         JR      NZ,BVER9         ;Go on error
2770 03         02750         INC     BC               ;Bump sector table ptr
2771 3E00       02760 BVER5   LD      A,0              ;P/u sector #
2773 3C         02770         INC     A                ;Bump it up
2774 327227     02780         LD      (BVER5+1),A      ;  and save new #
2777 5F         02790         LD      E,A              ;Xfer to sector register
2778 3AE82A     02800         LD      A,(SECCYL)       ;Is this = a cyl?
```

Format Execution Code

```
277B BB        Ø281Ø          CP      E
277C 28ØC      Ø282Ø          JR      Z,BVER8          ;Go if cyl done
277E 3AE92A    Ø283Ø          LD      A,(SECTRK)       ;Is this a track's worth?
2781 BB        Ø284Ø          CP      E
2782 2ØE3      Ø285Ø          JR      NZ,BVER3         ;Loop if not
2784 326927    Ø286Ø          LD      (BVER4+1),A      ;Update the add for side2
2787 Ø3        Ø287Ø          INC     BC
2788 18DA      Ø288Ø          JR      VERSKEW
               Ø289Ø ;
               Ø29ØØ ;        Readjust for end of cylinder
               Ø291Ø ;
278A 7A        Ø292Ø BVER8    LD      A,D              ;P/u current cyl position
278B 14        Ø293Ø          INC     D                ;Bump to next cyl
278C FDBEØ6    Ø294Ø          CP      (IY+6)           ;Cp to highest # cyl
278F Ø14727    Ø295Ø          LD      BC,BVER1         ;Go if more to verify
2792 2Ø9Ø      Ø296Ø          JR      NZ,CKWAIT        ;  after checking WAIT
               Ø297Ø ;
               Ø298Ø ;        Shift the FREE table to LOCKOUT table
               Ø299Ø ;
2794 21ØØ2E    Ø3ØØØ MOVFREE  LD      HL,GATBUF        ;Ptr to allocation info
2797 116Ø2E    Ø3Ø1Ø          LD      DE,GATBUF+6ØH    ;Lockout table
279A Ø6ØØ      Ø3Ø2Ø          LD      B,Ø
279C FD4EØ6    Ø3Ø3Ø          LD      C,(IY+6)         ;P/u hi cyl
279F ØC        Ø3Ø4Ø          INC     C                ;Offset from Ø
27AØ EDBØ      Ø3Ø5Ø          LDIR                     ;Shift info to the lockout tbl
27A2 ØEØD      Ø3Ø6Ø          LD      C,CR             ;Print a newline
27A4           Ø3Ø7Ø          @@DSP
27A4 3EØ2      ØØØ21          LD      A,2
27A6 EF        ØØØ22          RST     4Ø
27A7 185D      Ø3Ø8Ø          JR      CALCDIR          ;Go finish DIR init
               Ø3Ø9Ø ;
               Ø31ØØ ;        Got verify error
               Ø311Ø ;
27A9 FEØ5      Ø312Ø BVER9    CP      5                ;Data rec not found?
27AB 28Ø5      Ø313Ø          JR      Z,BVER1Ø
27AD FEØ4      Ø314Ø          CP      4                ;Parity error?
27AF C2A529    Ø315Ø          JP      NZ,IOERR         ;Quit on any other
27B2 D5        Ø316Ø BVER1Ø   PUSH    DE
27B3           Ø317Ø          @@DSPLY STAR$            ;Show the * lockout
               ØØØ23          IFEQ    Ø1H,1
27B3 21692C    ØØØ24          LD      HL,STAR$
               ØØØ25          ENDIF
27B6 3EØA      ØØØ26          LD      A,1Ø
27B8 EF        ØØØ27          RST     4Ø
27B9 D1        Ø318Ø          POP     DE
27BA 6A        Ø319Ø          LD      L,D              ;Pt to this cyl
27BB 262E      Ø32ØØ          LD      H,GATBUF<-8      ;  in the GAT
27BD 36FF      Ø321Ø          LD      (HL),ØFFH        ;Lockout this cylinder
27BF 18C9      Ø322Ø          JR      BVER8            ;Continue verifying
               Ø323Ø ;
               Ø324Ø ;        Hard drive format - most work done by controller
               Ø325Ø ;
27C1 218C39    Ø326Ø HRDRV    LD      HL,LASTMSG       ;Give one last chance to
27C4 FDCBØ35E  Ø327Ø          BIT     3,(IY+3)         ;  abort before wiping
27C8 28Ø9      Ø328Ø          JR      Z,AFLOP          ;  disk unless floppy
27CA CD5D2A    Ø329Ø          CALL    GET3             ;Is hard, get response
27CD 7E        Ø33ØØ          LD      A,(HL)           ;P/u 1st char of resp
27CE FE59      Ø331Ø          CP      'Y'              ;Must be yes to continue
27DØ C2B929    Ø332Ø          JP      NZ,FMTABT
```

Format Execution Code

```
27D3 3A1C26     Ø333Ø AFLOP   LD      A,(SYSPRM+1)     ;Bypass the formatting
27D6 B7         Ø334Ø         OR      A                ;  if system info only
27D7 2ØØC       Ø335Ø         JR      NZ,HRDRV1
27D9            Ø336Ø         @@DSPLY FMTG$            ;"formatting - be patient
                ØØØ28         IFEQ    Ø1H,1
27D9 216E2C     ØØØ29         LD      HL,FMTG$
                ØØØ3Ø         ENDIF
27DC 3EØA       ØØØ31         LD      A,1Ø
27DE EF         ØØØ32         RST     4Ø
27DF CD132A     Ø337Ø         CALL    FMTHD            ;Format hard drive
27E2 C2A529     Ø338Ø         JP      NZ,IOERR
27E5 FD7EØ7     Ø339Ø HRDRV1  LD      A,(IY+7)         ;# of sectors/gran
27E8 57         Ø34ØØ         LD      D,A              ;-> reg E
27E9 E61F       Ø341Ø         AND     1FH
27EB 5F         Ø342Ø         LD      E,A
27EC 1C         Ø343Ø         INC     E                ;Bump for Ø offset
27ED AA         Ø344Ø         XOR     D
27EE Ø7         Ø345Ø         RLCA                     ;Get # of heads
27EF Ø7         Ø346Ø         RLCA                     ;Into reg D
27FØ Ø7         Ø347Ø         RLCA
27F1 3C         Ø348Ø         INC     A                ;Adjust for zero offset
27F2 4F         Ø349Ø         LD      C,A
27F3            Ø35ØØ         @@MUL8                   ;Multiply E x C
27F3 3E5A       ØØØ33         LD      A,9Ø
27F5 EF         ØØØ34         RST     4Ø
27F6 FDCBØ46E   Ø351Ø         BIT     5,(IY+4)         ;2-sided?
27FA 28Ø1       Ø352Ø         JR      Z,$+3
27FC 87         Ø353Ø         ADD     A,A              ;Twice the number
27FD 32E82A     Ø354Ø         LD      (SECCYL),A
28ØØ FDCBØ35E   Ø355Ø         BIT     3,(IY+3)         ;Floppy?
28Ø4 288E       Ø356Ø         JR      Z,MOVFREE        ;Form lock table instead
                Ø357Ø ;
                Ø358Ø ;       Routine to calculate the directory cylinder
                Ø359Ø ;
28Ø6 CDFF29     Ø36ØØ CALCDIR CALL    RESTOR           ;Step in
28Ø9 C2A529     Ø361Ø         JP      NZ,IOERR         ;Go on error
28ØC 262E       Ø362Ø         LD      H,GATBUF<-8
28ØE FD6EØ6     Ø363Ø         LD      L,(IY+6)         ;P/u highest # cylinder
2811 Ø1ØØØØ     Ø364Ø DIRPARM LD      BC,ØØØØ          ;P/U 'DIR=' parm
2814 79         Ø365Ø         LD      A,C              ;Check if entered
2815 BØ         Ø366Ø         OR      B
2816 28Ø6       Ø367Ø         JR      Z,NODIR          ;Calc one if not entered
2818 BD         Ø368Ø         CP      L                ;Entered so check if
2819 3ØØ3       Ø369Ø         JR      NC,NODIR         ;  within cylinders
281B 6F         Ø37ØØ         LD      L,A              ;Is ok, use it
281C 18Ø3       Ø371Ø         JR      DIRSET
281E 2C         Ø372Ø NODIR   INC     L                ;Adj for zero offset
281F CB3D       Ø373Ø         SRL     L                ;Divide by 2 to find
2821 ØEØØ       Ø374Ø DIRSET  LD      C,Ø              ;  disk midpoint
                Ø375Ø ;
                Ø376Ø ;       Perform expanding binary search to find
                Ø377Ø ;       A cylinder available for the directory
                Ø378Ø ;
2823 7E         Ø379Ø CALC1   LD      A,(HL)           ;Is this cylinder
2824 3C         Ø38ØØ         INC     A                ;Available or locked out?
2825 2Ø19       Ø381Ø         JR      NZ,GENSYS        ;Bypass if available
2827 ØC         Ø382Ø         INC     C                ;Bump C
2828 79         Ø383Ø         LD      A,C
2829 ØF         Ø384Ø         RRCA                     ;Test if odd or even
```

Format Execution Code

```
282A 7D         03850          LD      A,L             ;Get current test pos
282B 3009       03860          JR      NC,CALC2        ;Jump if C was even
282D 81         03870          ADD     A,C             ;Add to previous pos
282E 6F         03880          LD      L,A
282F FDBE06     03890          CP      (IY+6)          ;Go over the top?
2832 20EF       03900          JR      NZ,CALC1        ;Loop if not
2834 1804       03910          JR      CALC3           ;Else abort
2836 91         03920 CALC2    SUB     C               ;Try a lower cylinder #
2837 6F         03930          LD      L,A
2838 20E9       03940          JR      NZ,CALC1        ;At cylinder 0?
283A 217C2C     03950 CALC3    LD      HL,NOCYL$       ;"no dir space avail...
283D C3B929     03960          JP      FMTABT
                03970 ;
                03980 ;        Generate the system initialization
                03990 ;
2840 FD7509     04000 GENSYS   LD      (IY+9),L        ;Stuff the dir cyl
2843 7D         04010          LD      A,L
2844 CD502A     04020          CALL    CVDEC           ;Cvrt reg A to 2 dec digs
2847 ED43C62C   04030          LD      (DIRASC$),BC    ;Stuff into the message
284B            04040          @@DSPLY DIRCYL$         ;"dir will be placed...
                00035          IFEQ    01H,1
284B 21A12C     00036          LD      HL,DIRCYL$
                00037          ENDIF
284E 3E0A       00038          LD      A,10
2850 EF         00039          RST     40
2851            04050          @@DSPLY IPLSYS$         ;"initializing...
                00040          IFEQ    01H,1
2851 21C92C     00041          LD      HL,IPLSYS$
                00042          ENDIF
2854 3E0A       00043          LD      A,10
2856 EF         00044          RST     40
2857 21002E     04060          LD      HL,GATBUF
285A 7E         04070          LD      A,(HL)          ;P/u GAT byte for 1st
285B F601       04080          OR      1               ;  cylinder & show 1st
285D 77         04090          LD      (HL),A          ;  gran in use for BOOTs
285E FD7E09     04100          LD      A,(IY+9)        ;Dir cyl # into DIR/SYS
2861 32D32A     04110          LD      (DIRDIR+16H),A
2864 6F         04120          LD      L,A             ;Show entire directory
2865 36FF       04130          LD      (HL),0FFH       ;  cylinder used
                04140 ;
                04150 ;        Update BOOT for DIR & step rate
                04160 ;
2867 FD7E09     04170          LD      A,(IY+9)        ;Dir cyl into BOOT
286A 32022F     04180          LD      (BOOT+2),A
286D 3A0026     04190          LD      A,(BOOTST$)     ;P/u offset
2870 6F         04200          LD      L,A
2871 262F       04210          LD      H,BOOT<-8
2873 3AE72A     04220          LD      A,(STEPDFT)     ;P/u boot step rate
                04230          IF      @MOD2
                04240          OR      80H             ;Create single byte opcod
                04250          ENDIF
2876 77         04260          LD      (HL),A          ;  & set into BOOT
2877 110000     04270          LD      DE,0            ;Init for cyl 0, sect 0
287A CD272A     04280          CALL    VERSEC          ;Test if formatted
287D 21352D     04290          LD      HL,NOTFMT$      ;"Can't, not formatted
2880 C2BC29     04300          JP      NZ,EXTERR       ;Error if not
2883 21002F     04310          LD      HL,BOOT         ;Pt to Data disk BOOT
2886 CD182A     04320          CALL    WRSEC           ;  & write it
2889 CC442A     04330          CALL    Z,WRDIR1        ;Verify after write
```

Format Execution Code

```
288C C2A529                JP      NZ,IOERR        ;  & display '.'
288F 11Ø1ØØ                LD      DE,1            ;Pt to cyl Ø, sector 1
2892 21ØØ2F                LD      HL,BOOT         ;Pt to the sector 1 boot
2895 CD182A                CALL    WRSEC           ;Write Ø/1
2898 CC442A                CALL    Z,WRDIR+3       ;Verify after write
289B C2A529                JP      NZ,IOERR
                  ;
                  ;        Complete GAT construction
                  ;
289E FD7EØ6                LD      A,(IY+6)        ;P/u highest # cylinder
28A1 D622                  SUB     22H             ;  & adj offset from 34
28A3 32CC2E                LD      (GATBUF+ØCCH),A ;Stuff GAT cyl excess
28A6 FD7EØ4                LD      A,(IY+4)        ;P/u # of sides
28A9 E6AØ                  AND     8ØH+2ØH
28AB 47                    LD      B,A             ;Save tempy in B
28AC FD7EØ3                LD      A,(IY+3)        ;P/u density
28AF E64Ø                  AND     4ØH             ;Mask it,
28B1 BØ                    OR      B               ;  merge in sides
28B2 47                    LD      B,A             ;  and save it
28B3 FD7EØ8                LD      A,(IY+8)        ;P/u # of grans/cyl
28B6 Ø7                    RLCA
28B7 Ø7                    RLCA                    ;  to bits Ø-2
28B8 Ø7                    RLCA
28B9 E6Ø7                  AND     7               ;Mask it
28BB 325229                LD      (CYLGRN+1),A
28BE BØ                    OR      B               ;Merge the two
28BF F68Ø                  OR      8ØH             ;Show it's a data disk
28C1 32CD2E                LD      (GATBUF+ØCDH),A ;Stuff into GAT
                  ;
28C4 11F52E                LD      DE,GATBUF+255-1Ø        ;6.2 Media Data Block
28C7 21DB28                LD      HL,LSIID        ;Point to header
28CA Ø1Ø4ØØ                LD      BC,Ø4           ;Set length &
28CD EDBØ                  LDIR                    ;  move it
28CF FDE5                  PUSH    IY              ;Get DCT address
28D1 E1                    POP     HL              ;  into HL
28D2 23                    INC     HL              ;Bypass the driver vector
28D3 23                    INC     HL
28D4 23                    INC     HL
28D5 ØEØ7                  LD      C,7             ;Bytes to move
28D7 EDBØ                  LDIR
28D9 18Ø4                  JR      WRGAT1          ;Skip around string
28DB Ø3           LSIID    DB      Ø3,'LSI'
     4C 53 49
                  ;
                  ;        Write copy of GAT into Ø/3
                  ;
                  WRGAT1
28DF 21ØØ2E                LD      HL,GATBUF       ;Pt to GAT buffer
28E2 16ØØ                  LD      D,Ø             ;Write it out to
28E4 1EØ3                  LD      E,3             ;Cyl Ø, sector 3
28E6 CD182A                CALL    WRSEC           ;Write Ø/3
28E9 CC442A                CALL    Z,WRDIR1        ;Verify after write
28EC C2A529                JP      NZ,IOERR        ;Quit on error
                  ;
                  ;        Write the system information sector
                  ;
28EF 21ØØ3Ø                LD      HL,HITBUF       ;Zero out buffer
28F2 36ØØ         GSYS1    LD      (HL),Ø
28F4 2C                    INC     L
```

Format Execution Code

```
28F5 2ØFB      Ø492Ø          JR      NZ,GSYS1
28F7 21ØØ3Ø    Ø493Ø          LD      HL,HITBUF         ;Set first byte to OSVER
28FA 3662      Ø494Ø          LD      (HL),RLS          ;  for release number
28FC 2E2Ø      Ø495Ø          LD      L,2ØH             ;Point hl to AUTO buffer
28FE 36ØD      Ø496Ø          LD      (HL),ØDH          ;Put in terminator
29ØØ 11Ø2ØØ    Ø497Ø          LD      DE,2              ;Pt to cyl Ø, sector 2
29Ø3 6A        Ø498Ø          LD      L,D               ;Hl now points to HITBUF
29Ø4 CD182A    Ø499Ø          CALL    WRSEC             ;Write Ø/2
29Ø7 CC442A    Ø5ØØØ          CALL    Z,WRDIR1          ;Verify after write
29ØA C2A529    Ø5Ø1Ø          JP      NZ,IOERR          ;Quit on error
29ØD 2E2Ø      Ø5Ø2Ø          LD      L,2ØH             ;Zero this out for use
29ØF 36ØØ      Ø5Ø3Ø          LD      (HL),Ø            ;  when writing HIT
               Ø5Ø4Ø ;
               Ø5Ø5Ø ;        Write out the directory GAT
               Ø5Ø6Ø ;
2911 21ØØ2E    Ø5Ø7Ø          LD      HL,GATBUF         ;Pt to GAT sector buffer
2914 FD56Ø9    Ø5Ø8Ø          LD      D,(IY+9)          ;P/u the dir cyl
2917 5D        Ø5Ø9Ø          LD      E,L               ;Denote sector Ø
2918 CD412A    Ø51ØØ          CALL    WRDIR             ;Write the GAT
291B C2A529    Ø511Ø          JP      NZ,IOERR
               Ø512Ø ;
               Ø513Ø ;        Construct the HIT
               Ø514Ø ;
291E 21ØØ3Ø    Ø515Ø          LD      HL,HITBUF         ;Point to the HIT buffer
2921 36A2      Ø516Ø          LD      (HL),ØA2H         ;Stuff BOOT/SYS hash code
2923 23        Ø517Ø          INC     HL
2924 36C4      Ø518Ø          LD      (HL),ØC4H         ;Stuff DIR/SYS hash code
2926 2B        Ø519Ø          DEC     HL
2927 FD56Ø9    Ø52ØØ          LD      D,(IY+9)          ;P/u dir cyl #
292A 1EØ1      Ø521Ø          LD      E,1               ;Pt to sector 1
292C CD412A    Ø522Ø          CALL    WRDIR             ;Write the HIT
292F C2A529    Ø523Ø          JP      NZ,IOERR
2932 11ØØ3Ø    Ø524Ø          LD      DE,HITBUF         ;Establish buffer for
2935 219D2A    Ø525Ø          LD      HL,BOOTDIR        ;  dir records
2938 Ø12ØØØ    Ø526Ø          LD      BC,32             ;Move BOOT/SYS dir record
293B EDBØ      Ø527Ø          LDIR                      ;  into 1st slot
293D FD56Ø9    Ø528Ø          LD      D,(IY+9)          ;P/u dir cyl
294Ø 1EØ2      Ø529Ø          LD      E,2               ;This will be sector 2
2942 21ØØ3Ø    Ø53ØØ          LD      HL,HITBUF         ;Pt to buffer start
2945 CD412A    Ø531Ø          CALL    WRDIR             ;Write the sector
2948 C2A529    Ø532Ø          JP      NZ,IOERR
294B 3AE82A    Ø533Ø          LD      A,(SECCYL)        ;P/u # of records
294E 32D12A    Ø534Ø          LD      (DIRDIR+14H),A    ;  & stuff into DIR/SYS
2951 3EØØ      Ø535Ø CYLGRN   LD      A,Ø               ;P/u # grans/cyl
2953 FDCBØ46E  Ø536Ø          BIT     5,(IY+4)          ;Test 2-sided
2957 28Ø2      Ø537Ø          JR      Z,$+4
2959 87        Ø538Ø          ADD     A,A               ;Double count on 2-sided
295A 3C        Ø539Ø          INC     A                 ;Plus 1 for Ø offset adj
295B 32D42A    Ø54ØØ          LD      (DIRDIR+17H),A    ;Stuf in DIR/SYS
295E FD7EØ9    Ø541Ø          LD      A,(IY+9)          ;P/u the dir cyl # &
2961 32D32A    Ø542Ø          LD      (DIRDIR+16H),A    ;  stuff into the DIR rec
2964 21BD2A    Ø543Ø          LD      HL,DIRDIR         ;Pt to start of DIR data
2967 11ØØ3Ø    Ø544Ø          LD      DE,HITBUF         ;Pt to start of dir buf
296A Ø12ØØØ    Ø545Ø          LD      BC,32             ;Move DIR/SYS into buf
296D EDBØ      Ø546Ø          LDIR
296F FD56Ø9    Ø547Ø          LD      D,(IY+9)          ;P/u dir cyl #
2972 1EØ3      Ø548Ø          LD      E,3               ;Write as sector 3
2974 21ØØ3Ø    Ø549Ø          LD      HL,HITBUF         ;Pt to start of buffer
2977 CD412A    Ø55ØØ          CALL    WRDIR             ;Write the sector
```

Format Execution Code

```
297A 2Ø29       Ø551Ø          JR      NZ,IOERR
297C 21ØØ3Ø     Ø552Ø          LD      HL,HITBUF          ;Zero the 1st 32 bytes
297F Ø62Ø       Ø553Ø          LD      B,32               ;  of the buffer to clear
2981 36ØØ       Ø554Ø GSYS2    LD      (HL),Ø             ;Where we stuffed the
2983 23         Ø555Ø          INC     HL                 ;  BOOT & DIR dir records
2984 1ØFB       Ø556Ø          DJNZ    GSYS2
2986 FD56Ø9     Ø557Ø          LD      D,(IY+9)           ;P/u dir cyl #
2989 1EØ4       Ø558Ø          LD      E,4                ;Cont writing at sect 4
298B 21ØØ3Ø     Ø559Ø GSYS3    LD      HL,HITBUF          ;Pt to start of buffer
298E CD412A     Ø56ØØ          CALL    WRDIR              ;Write the sector
2991 2Ø12       Ø561Ø          JR      NZ,IOERR
                Ø562Ø ;
                Ø563Ø ;        Write the remaining directory
                Ø564Ø ;
2993 1C         Ø565Ø          INC     E                  ;Bump the sector pointer
2994 3AE82A     Ø566Ø          LD      A,(SECCYL)         ;P/u highest # sector
2997 BB         Ø567Ø          CP      E                  ;Are we finished yet?
2998 2ØF1       Ø568Ø          JR      NZ,GSYS3           ;Loop if not
299A CDDF29     Ø569Ø          CALL    EXIT2              ;Get system disk
299D            Ø57ØØ          @@DSPLY FMTCAO$            ;"formatting complete...
                ØØØ45          IFEQ    Ø1H,1
299D 21ØE2D     ØØØ46          LD      HL,FMTCAO$
                ØØØ47          ENDIF
29AØ 3EØA       ØØØ48          LD      A,1Ø
29A2 EF         ØØØ49          RST     4Ø
29A3 1823       Ø571Ø          JR      EXIT
                Ø572Ø ;
                Ø573Ø ;        Exit procedures
                Ø574Ø ;
29A5 F5         Ø575Ø IOERR    PUSH    AF                 ;Save errcod
29A6 CDDF29     Ø576Ø          CALL    EXIT2              ;Interrupts on if needed
29A9 F1         Ø577Ø          POP     AF                 ;Rcvr errcod
29AA FE3F       Ø578Ø          CP      63                 ;Extended errror?
29AC 28ØE       Ø579Ø          JR      Z,EXTERR           ;Go if so
29AE 6F         Ø58ØØ          LD      L,A                ;Error code to HL
29AF 26ØØ       Ø581Ø          LD      H,Ø
29B1 F6CØ       Ø582Ø          OR      ØCØH               ;Mask to ABORT with brief
29B3 4F         Ø583Ø          LD      C,A                ;Error code to C
29B4            Ø584Ø          @@ERROR                    ;  for error display
29B4 3E1A       ØØØ5Ø          LD      A,26
29B6 EF         ØØØ51          RST     4Ø
29B7 18ØC       Ø585Ø          JR      ERREXIT
                Ø586Ø ;
29B9            Ø587Ø BREAK    EQU     $
29B9 21242D     Ø588Ø FMTABT   LD      HL,FMTABT$         ;"Command aborted
29BC            Ø589Ø EXTERR   @@LOGOT                    ;Some error to abort job
                ØØØ52          IFEQ    ØØH,1
                ØØØ53          LD      HL,
                ØØØ54          ENDIF
29BC 3EØC       ØØØ55          LD      A,12
29BE EF         ØØØ56          RST     4Ø
29BF CDDF29     Ø59ØØ          CALL    EXIT2              ;Get system disk
29C2 21FFFF     Ø591Ø          LD      HL,-1              ;Set abort code
29C5 22C929     Ø592Ø ERREXIT  LD      (RETCOD),HL
29C8 21ØØØØ     Ø593Ø EXIT     LD      HL,Ø               ;Init to no error
29C9            Ø594Ø RETCOD   EQU     $-2
29CB E5         Ø595Ø          PUSH    HL
29CC FDE5       Ø596Ø          PUSH    IY                 ;Transfer the saved
29CE D1         Ø597Ø          POP     DE                 ;  system DCT back
```

Format Execution Code

```
29CF 21DD2A     Ø598Ø           LD      HL,SYSDCT         ;  into the system
29D2 Ø1ØAØØ     Ø599Ø           LD      BC,1Ø             ;  DCT slot
29D5 EDBØ       Ø6ØØØ           LDIR
29D7 E1         Ø6Ø1Ø           POP     HL
29D8 31ØØØØ     Ø6Ø2Ø SPSAV     LD      SP,$-$            ;P/u the stack pointer
29DB            Ø6Ø3Ø           @@CKBRKC                  ;Clear break bit
29DB 3E6A       ØØØ57           LD      A,1Ø6
29DD EF         ØØØ58           RST     4Ø
29DE C9         Ø6Ø4Ø           RET                       ;  & exit to caller
                Ø6Ø5Ø ;
29DF 3A2B2A     Ø6Ø6Ø EXIT2     LD      A,(FMTDRV+1)      ;P/u drive # just fmtd
29E2 3C         Ø6Ø7Ø           INC     A                 ;If drive never entered,
29E3 C8         Ø6Ø8Ø           RET     Z                 ;  just return
29E4 3D         Ø6Ø9Ø           DEC     A                 ;If Ø, we need a system
29E5 2ØØD       Ø61ØØ           JR      NZ,EXIT4
29E7 21EF2C     Ø611Ø           LD      HL,PMTSYS$        ;"load system disk...
29EA            Ø612Ø           @@DSPLY
                ØØØ59           IFEQ    ØØH,1
                ØØØ6Ø           LD      HL,
                ØØØ61           ENDIF
29EA 3EØA       ØØØ62           LD      A,1Ø
29EC EF         ØØØ63           RST     4Ø
29ED            Ø613Ø EXIT3     @@KEY                     ;Request a key
29ED 3EØ1       ØØØ64           LD      A,1
29EF EF         ØØØ65           RST     4Ø
29FØ FEØD       Ø614Ø           CP      CR                ;Must be <ENTER>
29F2 2ØF9       Ø615Ø           JR      NZ,EXIT3
29F4 18Ø9       Ø616Ø EXIT4     JR      RESTOR            ;Restore disk to cyl Ø
                Ø617Ø ;
                Ø618Ø ;         Disk I/O requests
                Ø619Ø ;
29F6 C5         Ø62ØØ DRVNOP    PUSH    BC
29F7 AF         Ø621Ø           XOR     A
29F8 183Ø       Ø622Ø           JR      FMTDRV
29FA C5         Ø623Ø SELECT    PUSH    BC
29FB 3EØ1       Ø624Ø           LD      A,1
29FD 182B       Ø625Ø           JR      FMTDRV
29FF C5         Ø626Ø RESTOR    PUSH    BC
2AØØ 3EØ4       Ø627Ø           LD      A,4
2AØ2 1826       Ø628Ø           JR      FMTDRV
2AØ4 C5         Ø629Ø STEPIN    PUSH    BC
2AØ5 3EØ5       Ø63ØØ           LD      A,5
2AØ7 1821       Ø631Ø           JR      FMTDRV
2AØ9 C5         Ø632Ø RSELCT    PUSH    BC
2AØA 3EØ7       Ø633Ø           LD      A,7
2AØC 181C       Ø634Ø           JR      FMTDRV
2AØE C5         Ø635Ø WRCYL     PUSH    BC
2AØF 3EØF       Ø636Ø           LD      A,15
2A11 1817       Ø637Ø           JR      FMTDRV
2A13 C5         Ø638Ø FMTHD     PUSH    BC
2A14 3EØC       Ø639Ø           LD      A,12
2A16 1812       Ø64ØØ           JR      FMTDRV
2A18 C5         Ø641Ø WRSEC     PUSH    BC
2A19 3EØD       Ø642Ø           LD      A,13
2A1B 18ØD       Ø643Ø           JR      FMTDRV
2A1D C5         Ø644Ø WRSYS     PUSH    BC
2A1E 3EØE       Ø645Ø           LD      A,14
2A2Ø 18Ø8       Ø646Ø           JR      FMTDRV
2A22 C5         Ø647Ø RDSEC     PUSH    BC
```

Format Execution Code

```
2A23 3EØ9       Ø648Ø          LD      A,9
2A25 18Ø3       Ø649Ø          JR      FMTDRV
2A27 C5         Ø65ØØ VERSEC   PUSH    BC
2A28 3EØA       Ø651Ø          LD      A,1Ø
2A2A ØEFF       Ø652Ø FMTDRV   LD      C,-1             ;P/u drive #
2A2C C628       Ø653Ø          ADD     A,4Ø             ;Adjust SVC #
2A2E EF         Ø654Ø          RST     4Ø
2A2F C1         Ø655Ø          POP     BC
2A3Ø C9         Ø656Ø          RET
                Ø657Ø ;
                Ø658Ø ;        Perform a verification to ensure system sector
                Ø659Ø ;
2A31 CD272A     Ø66ØØ VERSYS   CALL    VERSEC           ;Sector verify
2A34 28Ø6       Ø661Ø          JR      Z,VERS1          ;Bypass if not system
2A36 D6Ø6       Ø662Ø          SUB     6                ;Test read system retcod
2A38 C8         Ø663Ø          RET     Z                ;Go if that's what it was
2A39 C6Ø6       Ø664Ø          ADD     A,6              ;Restore orig retcod
2A3B C9         Ø665Ø          RET
2A3C F6Ø1       Ø666Ø VERS1    OR      1                ;S/b system, found data
2A3E 3EØØ       Ø667Ø          LD      A,Ø
2A4Ø C9         Ø668Ø          RET
                Ø669Ø ;
2A41 CD1D2A     Ø67ØØ WRDIR    CALL    WRSYS            ;Write the DIR sector
2A44 C4312A     Ø671Ø WRDIR1   CALL    NZ,VERSYS        ;Verify after write
2A47 CØ         Ø672Ø          RET     NZ
2A48 D5         Ø673Ø          PUSH    DE
2A49 ØE2E       Ø674Ø          LD      C,'.'            ;Display a period
2A4B            Ø675Ø          @@DSP                    ;  for every sector written
2A4B 3EØ2       ØØØ66          LD      A,2
2A4D EF         ØØØ67          RST     4Ø
2A4E D1         Ø676Ø          POP     DE
2A4F C9         Ø677Ø          RET
                Ø678Ø ;
                Ø679Ø ;        Routine to convert reg A to 2 decimal digits
                Ø68ØØ ;
2A5Ø ØE3Ø       Ø681Ø CVDEC    LD      C,3ØH            ;Init msd to Ø
2A52 D6ØA       Ø682Ø CVD1     SUB     1Ø               ;Sub 1Ø until underflow
2A54 38Ø3       Ø683Ø          JR      C,CVD2
2A56 ØC         Ø684Ø          INC     C                ;Inc the count
2A57 18F9       Ø685Ø          JR      CVD1
2A59 C63A       Ø686Ø CVD2     ADD     A,3AH            ;Add back 1Ø + 'Ø'
2A5B 47         Ø687Ø          LD      B,A              ;Lsd to B
2A5C C9         Ø688Ø          RET
                Ø689Ø ;
                Ø69ØØ ;        Routines to convert input strings to UC
                Ø691Ø ;        HL => Prompt string
                Ø692Ø ;
2A5D            Ø693Ø GET3     @@DSPLY                  ;Display the prompt
                ØØØ68          IFEQ    ØØH,1
                ØØØ69          LD      HL,
                ØØØ7Ø          ENDIF
2A5D 3EØA       ØØØ71          LD      A,1Ø
2A5F EF         ØØØ72          RST     4Ø
2A6Ø Ø1ØØØ3     Ø694Ø          LD      BC,3<8           ;Init 3 keys max
2A63 18Ø3       Ø695Ø          JR      $+5
2A65 Ø1ØØØ8     Ø696Ø GET8     LD      BC,8<8           ;8-chars max
2A68 21ØØ3Ø     Ø697Ø          LD      HL,HITBUF        ;Buffer area
2A6B            Ø698Ø GET8A    @@KEYIN                  ;Enter them
2A6B 3EØ9       ØØØ73          LD      A,9
```

Format Execution Code

```
2A6D EF         00074          RST     40
2A6E DAB929     06990          JP      C,FMTABT        ;Quit on Break
2A71 78         07000          LD      A,B             ;Get length of response
2A72 B7         07010          OR      A
2A73 C8         07020          RET     Z               ;Back if Enter only
                07030 ;
                07040 ;        Routine to convert n-character string to UC
                07050 ;
2A74 F5         07060          PUSH    AF              ;Save the registers
2A75 C5         07070          PUSH    BC
2A76 E5         07080          PUSH    HL
2A77 7E         07090 GETUC    LD      A,(HL)          ;P/u a char
2A78 FE61       07100          CP      'a'             ;Skip if below 'a'
2A7A 3806       07110          JR      C,GETUC1
2A7C FE7B       07120          CP      'z'+1           ;  or above 'z'
2A7E 3002       07130          JR      NC,GETUC1
2A80 CBAE       07140          RES     5,(HL)          ;  else convert to UC
2A82 23         07150 GETUC1   INC     HL              ;Bump the buffer ptr
2A83 10F2       07160          DJNZ    GETUC           ;Loop thru all chars
2A85 E1         07170          POP     HL
2A86 C1         07180          POP     BC
2A87 F1         07190          POP     AF
2A88 C9         07200          RET
                07210 ;
                07220 ;        Routine to display the cylinder number
                07230 ;
2A89 C5         07240 DSPCYL   PUSH    BC              ;Save ASCII cylinder #
2A8A 0E08       07250          LD      C,8             ;Back up twice &
2A8C            07260          @@DSP                   ;  output new position
2A8C 3E02       00075          LD      A,2
2A8E EF         00076          RST     40
2A8F 0E08       07270          LD      C,8
2A91            07280          @@DSP
2A91 3E02       00077          LD      A,2
2A93 EF         00078          RST     40
2A94 C1         07290          POP     BC              ;Recover cyl #
2A95            07300          @@DSP                   ;Send MSD
2A95 3E02       00079          LD      A,2
2A97 EF         00080          RST     40
2A98 48         07310          LD      C,B
2A99            07320          @@DSP                   ;Send LSD
2A99 3E02       00081          LD      A,2
2A9B EF         00082          RST     40
2A9C C9         07330          RET
                07340 ;
                07350 ;        Formatting data and tables
                07360 ;
2A9D 5E         07370 BOOTDIR  DB      5EH,0,0,0,0,'BOOT    SYS',0F6H,37H
     00 00 00 00 42 4F 4F 54
     20 20 20 20 53 59 53 F6
     37
2AAF F5         07380          DB      0F5H,9CH,5,0,0,0,0FFH,0FFH,-1,-1,-1,-1,-1,-1
     9C 05 00 00 00 FF FF FF
     FF FF FF FF FF
2ABD 5D         07390 DIRDIR   DB      5DH,0,0,0,0,'DIR     SYS',0F6H,37H
     00 00 00 00 44 49 52 20
     20 20 20 20 53 59 53 F6
     37
2ACF 96         07400          DB      96H,42H,10,0,11H,1,0FFH,0FFH,0,0,0,0,0,0
```

Format Execution Code

```
     42 ØA ØØ 11 Ø1 FF FF ØØ
     ØØ ØØ ØØ ØØ ØØ
ØØØA          Ø741Ø SYSDCT  DS      1Ø
2AE7 ØØ       Ø742Ø STEPDFT DB      Ø               ;Boot step rate default
ØØØ1          Ø743Ø SECCYL  DS      1               ;# of sectors per cyl
ØØØ1          Ø744Ø SECTRK  DS      1               ;# of sectors per trk
              Ø745Ø ;
              Ø746Ø ;       Single density 5" format table
              Ø747Ø ;
2AEA ØA       Ø748Ø S5TBL   DB      1Ø,7
     Ø7
2AEC ØØ       Ø749Ø         DB      Ø,5,1,6,2,7,3,8,4,9
     Ø5 Ø1 Ø6 Ø2 Ø7 Ø3 Ø8 Ø4
     Ø9
2AF6 F6       Ø75ØØ         DB      -1Ø,-1Ø,-1Ø,-1Ø,-1Ø,-1Ø,-1Ø,-1Ø,14,ØFFH
     F6 F6 F6 F6 F6 F6 F6 ØE
     FF
2BØØ F1       Ø751Ø         DB      ØF1H,6,Ø,1,ØFEH
     Ø6 ØØ Ø1 FE
2BØ5 F3       Ø752Ø         DB      ØF3H,3,Ø,1,1,1,ØF7H,1,ØFFH,11,ØFFH
     Ø3 ØØ Ø1 Ø1 Ø1 F7 Ø1 FF
     ØB FF
2B1Ø Ø6       Ø753Ø         DB      6,Ø,1,ØFBH,Ø,ØE5H,1,ØF7H,1,ØFFH,13,ØFFH
     ØØ Ø1 FB ØØ E5 Ø1 F7 Ø1
     FF ØD FF
2B1C F2       Ø754Ø         DB      ØF2H,47H,ØFFH,ØF4H
     47 FF F4
2B2Ø ØØ       Ø755Ø         DB      Ø,1,2,3,4,5,6,7,8,9
     Ø1 Ø2 Ø3 Ø4 Ø5 Ø6 Ø7 Ø8
     Ø9
              Ø756Ø ;
              Ø757Ø ;       Double density 5" format table
              Ø758Ø ;
2B2A 12       Ø759Ø D5TBL   DB      18,1Ø
     ØA
2B2C ØØ       Ø76ØØ         DB      Ø,9,1,1Ø,2,11,3,12,4
     Ø9 Ø1 ØA Ø2 ØB Ø3 ØC Ø4
2B35 ØD       Ø761Ø         DB      13,5,14,6,15,7,16,8,17
     Ø5 ØE Ø6 ØF Ø7 1Ø Ø8 11
2B3E EE       Ø762Ø         DC      11,-18
     EE EE EE EE EE EE EE EE
     EE EE
2B49 14       Ø763Ø         DB      2Ø,4EH
     4E
2B4B F1       Ø764Ø         DB      ØF1H,12,Ø,3,ØF5H,1,ØFEH
     ØC ØØ Ø3 F5 Ø1 FE
2B52 F3       Ø765Ø         DB      ØF3H,3,Ø,1,1,1,ØF7H,22,4EH,12,Ø,3,ØF5H
     Ø3 ØØ Ø1 Ø1 Ø1 F7 16 4E
     ØC ØØ Ø3 F5
2B5F Ø1       Ø766Ø         DB      1,ØFBH,ØF5H,128,6DH,ØB6H
     FB F5 8Ø 6D B6
2B65 Ø1       Ø767Ø         DB      1,ØF7H,1,ØFFH,17,Ø4EH
     F7 Ø1 FF 11 4E
2B6B F2       Ø768Ø         DB      ØF2H,182,4EH,ØF4H
     B6 4E F4
2B6F ØØ       Ø769Ø         DB      Ø,1,2,3,4,5,6,7,8,9
     Ø1 Ø2 Ø3 Ø4 Ø5 Ø6 Ø7 Ø8
     Ø9
2B79 ØA       Ø77ØØ         DB      1Ø,11,12,13,14,15,16,17
```

Format Execution Code

```
     ØB ØC ØD ØE ØF 1Ø 11
               Ø771Ø ;
               Ø772Ø ;         Single density 8" format table
               Ø773Ø ;
2B81 1Ø        Ø774Ø S8TBL     DB      16,2
     Ø2
2B83 ØA        Ø775Ø           DB      1Ø,5,Ø,11,6,1,12,7,2,13,8,3,14,9,4,15
     Ø5 ØØ ØB Ø6 Ø1 ØC Ø7 Ø2
     ØD Ø8 Ø3 ØE Ø9 Ø4 ØF
2B93 FØ        Ø776Ø           DB      -16,-16,-16,28H,ØFFH
     FØ FØ 28 FF
2B98 F1        Ø777Ø           DB      ØF1H,6,Ø,1,ØFEH
     Ø6 ØØ Ø1 FE
2B9D F3        Ø778Ø           DB      ØF3H,3,Ø,1,1,1,ØF7H,11,ØFFH,6,Ø,1,ØFBH
     Ø3 ØØ Ø1 Ø1 Ø1 F7 ØB FF
     Ø6 ØØ Ø1 FB
2BAA ØØ        Ø779Ø           DB      Ø,ØE5H,1,ØF7H,1,ØFFH,2Ø,ØFFH
     E5 Ø1 F7 Ø1 FF 14 FF
2BB2 F2        Ø78ØØ           DB      ØF2H,2Ø8,ØFFH,ØF4H
     DØ FF F4
2BB6 ØA        Ø781Ø           DB      1Ø,Ø,6,12,2,8,14,4,5,11,1,7,13,3,9,15
     ØØ Ø6 ØC Ø2 Ø8 ØE Ø4 Ø5
     ØB Ø1 Ø7 ØD Ø3 Ø9 ØF
               Ø782Ø ;
               Ø783Ø ;         Double density 8" format table
               Ø784Ø ;
2BC6 1E        Ø785Ø D8TBL     DB      3Ø,12
     ØC
2BC8 ØØ        Ø786Ø           DB      Ø,1Ø,2Ø,1,11,21,2,12,22,3,13,23,4,14,24
     ØA 14 Ø1 ØB 15 Ø2 ØC 16
     Ø3 ØD 17 Ø4 ØE 18
2BD7 Ø5        Ø787Ø           DB      5,15,25,6,16,26,7,17,27,8,18,28,9,19,29
     ØF 19 Ø6 1Ø 1A Ø7 11 1B
     Ø8 12 1C Ø9 13 1D
2BE6 E2        Ø788Ø           DC      13,-3Ø
     E2 E2 E2 E2 E2 E2 E2 E2
     E2 E2 E2 E2
2BF3 14        Ø789Ø           DB      2Ø,4EH
     4E
2BF5 F1        Ø79ØØ           DB      ØF1H,ØCH,Ø,3,ØF5H,1,ØFEH
     ØC ØØ Ø3 F5 Ø1 FE
2BFC F3        Ø791Ø           DB      ØF3H,3,Ø,1,1,1,ØF7H,22,4EH,12,Ø,3,ØF5H
     Ø3 ØØ Ø1 Ø1 Ø1 F7 16 4E
     ØC ØØ Ø3 F5
2CØ9 Ø1        Ø792Ø           DB      1,ØFBH,ØF5H,128,6DH,ØB6H
     FB F5 8Ø 6D B6
2CØF Ø1        Ø793Ø           DB      1,ØF7H,1,ØFFH,17,4EH
     F7 Ø1 FF 11 4E
2C15 F2        Ø794Ø           DB      ØF2H,Ø,4EH,61,4EH,ØF4H
     ØØ 4E 3D 4E F4
2C1B ØØ        Ø795Ø           DB      Ø,2Ø,11,2,22,13,4,24,15,6,26,17,8,28,19
     14 ØB Ø2 16 ØD Ø4 18 ØF
     Ø6 1A 11 Ø8 1C 13
2C2A ØA        Ø796Ø           DB      1Ø,1,21,12,3,23,14,5,25,16,7,27,18,9,29
     Ø1 15 ØC Ø3 17 ØE Ø5 19
     1Ø Ø7 1B 12 Ø9 1D
               Ø797Ø ;
2C39 1D        Ø798Ø FMTCYL$ DB        29,'Formatting cylinder   ',3
     46 6F 72 6D 61 74 74 69
```

Format Execution Code

```
     6E 67 2Ø 63 79 6C 69 6E
     64 65 72 2Ø 2Ø 2Ø Ø3
2C51 1D        Ø799Ø VERCYL$ DB      29,'Verifying  cylinder   ',3
     56 65 72 69 66 79 69 6E
     67 2Ø 2Ø 63 79 6C 69 6E
     64 65 72 2Ø 2Ø 2Ø Ø3
2C69 2A        Ø8ØØØ STAR$   DB      '*   ',3
     2Ø 2Ø 2Ø Ø3
2C6E 46        Ø8Ø1Ø FMTG$   DB      'Formatting...',CR
     6F 72 6D 61 74 74 69 6E
     67 2E 2E 2E ØD
2C7C 4E        Ø8Ø2Ø NOCYL$  DB      'No cylinders available for directory',CR
     6F 2Ø 63 79 6C 69 6E 64
     65 72 73 2Ø 61 76 61 69
     6C 61 62 6C 65 2Ø 66 6F
     72 2Ø 64 69 72 65 63 74
     6F 72 79 ØD
2CA1 44        Ø8Ø3Ø DIRCYL$ DB      'Directory will be placed on cylinder '
     69 72 65 63 74 6F 72 79
     2Ø 77 69 6C 6C 2Ø 62 65
     2Ø 7Ø 6C 61 63 65 64 2Ø
     6F 6E 2Ø 63 79 6C 69 6E
     64 65 72 2Ø
2CC6 3Ø        Ø8Ø4Ø DIRASC$ DB      'ØØ',CR
     3Ø ØD
2CC9 ØA        Ø8Ø5Ø IPLSYS$ DB      LF,'Initializing DIRECTORY information: ',3
     49 6E 69 74 69 61 6C 69
     7A 69 6E 67 2Ø 44 49 52
     45 43 54 4F 52 59 2Ø 69
     6E 66 6F 72 6D 61 74 69
     6F 6E 3A 2Ø Ø3
2CEF ØA        Ø8Ø6Ø PMTSYS$ DB      LF,'Load SYSTEM diskette  <ENTER>',CR
     4C 6F 61 64 2Ø 53 59 53
     54 45 4D 2Ø 64 69 73 6B
     65 74 74 65 2Ø 2Ø 3C 45
     4E 54 45 52 3E ØD
2DØE ØA        Ø8Ø7Ø FMTCAO$ DB      LF,LF,'Formatting complete',CR
     ØA 46 6F 72 6D 61 74 74
     69 6E 67 2Ø 63 6F 6D 7Ø
     6C 65 74 65 ØD
2D24 ØA        Ø8Ø8Ø FMTABT$ DB      LF,'Command aborted',CR
     43 6F 6D 6D 61 6E 64 2Ø
     61 62 6F 72 74 65 64 ØD
2D35 ØA        Ø8Ø9Ø NOTFMT$ DB      LF,'Can''t, Diskette not formatted',CR
     43 61 6E 27 74 2C 2Ø 44
     69 73 6B 65 74 74 65 2Ø
     6E 6F 74 2Ø 66 6F 72 6D
     61 74 74 65 64 ØD
               Ø81ØØ ;
               Ø811Ø ;       Patch area
               Ø812Ø ;
2EØØ           Ø813Ø         ORG     $<-8+1<+8
ØØCB           Ø814Ø GATBUF  DS      2Ø3              ;GAT sector buffer
2ECB 62        Ø815Ø         DB      RLS,Ø,Ø,Ø,Ø      ;Ver, cyl exc, type, pswd
     ØØ ØØ ØØ ØØ
2EDØ 2Ø        Ø816Ø         DB      '        MM/DD/YY'
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 4D
     4D 2F 44 44 2F 59 59
2EEØ ØØ        Ø817Ø         DC      32,Ø
```

Format Execution Code

```
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
2FØØ           Ø818Ø BOOT    EQU     $
2FØØ           Ø819Ø CORE$   DEFL    $
43ØØ           Ø82ØØ         ORG     43ØØH            ;Execute at ROM BOOT
43ØØ           Ø821Ø         LORG    CORE$            ;  but load here
43ØØ ØØ        Ø822Ø         NOP
43Ø1 FEØØ      Ø823Ø         CP      Ø                ;P/u the DIR track
               Ø824Ø ;
               Ø825Ø         IF      @MOD2
               Ø826Ø         DB      83H
               Ø827Ø         ENDIF
               Ø828Ø         IF      @MOD4
43Ø3 F3        Ø829Ø         DI
               Ø83ØØ         ENDIF
43Ø4 ØØ        Ø831Ø         DC      12,Ø
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ
431Ø 118DFB    Ø832Ø         LD      DE,8Ø*11+CRT4+29         ;Mod 4 video locn
4313 212943    Ø833Ø         LD      HL,DATADSK$      ;Xfer error to vidmem
4316 Ø117ØØ    Ø834Ø         LD      BC,STRLEN
4319 EDBØ      Ø835Ø         LDIR
431B 11153E    Ø836Ø         LD      DE,64*8+CRT3+21 ;Mod 3 video locn
431E 212943    Ø837Ø         LD      HL,DATADSK$      ;Xfer error to vidmem
4321 Ø117ØØ    Ø838Ø         LD      BC,STRLEN
4324 EDBØ      Ø839Ø         LDIR
4326 C32643    Ø84ØØ STOP    JP      STOP
4329 43        Ø841Ø DATADSK$        DB      'Cannot boot, DATA DISK!'
     61 6E 6E 6F 74 2Ø 62 6F
     6F 74 2C 2Ø 44 41 54 41
     2Ø 44 49 53 4B 21
ØØ17           Ø842Ø STRLEN  EQU     $-DATADSK$
434Ø ØØ        Ø843Ø         DC      -$&ØFFH,Ø
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

Format Execution Code

```
44ØØ          Ø844Ø SAFESP  EQU     $
3ØØØ          Ø845Ø         ORG     CORE$+256
3ØØØ          Ø846Ø         LORG    CORE$+256
Ø1ØØ          Ø847Ø HITBUF  DS      256
              Ø848Ø ;
31ØØ          Ø849Ø         SUBTTL  '<Format Init Code>'
```

Format Init Code

```
3100            *GET    FORMAT2:3
                ;FORMAT2/ASM - Format Initialization Code
                ;
                ;       FORMAT routine entry point
                ;
                FORMAT
3100                    @@CKBRKC                ;Check for break
3100 3E6A               LD      A,106
3102 EF                 RST     40
3103 2804               JR      Z,FORMATA       ;Continue if no break
3105 21FFFF             LD      HL,-1           ;  else abort
3108 C9                 RET
                ;
3109 ED73D929   FORMATA LD      (SPSAV+1),SP    ;Save the stack pointer
310D E5                 PUSH    HL              ;Save cmdline ptr
310E                    @@DSPLY HELLO$          ;Hello message
                        IFEQ    01H,1
310E 21C236             LD      HL,HELLO$
                        ENDIF
3111 3E0A               LD      A,10
3113 EF                 RST     40
3114 CD5436             CALL    GETSYS2         ;Load SYS2 overlay
                ;
                ;       Read config sector & extract DCT # cyls
                ;
                        IF      @MOD4
3117 110200             LD      DE,2            ;Track 0, sector 2
311A 4A                 LD      C,D             ;Drive 0
                        ENDIF
                ;
                        IF      @MOD2
                        LD      C,0             ;Drive 0
311B                    @@GTDCT                 ;Fetch DCT
                        LD      A,81
                        RST     40
                        LD      A,(IY+3)        ;Get dct data
                        AND     28H             ;Bit 5/3
                        CP      20H             ;8" floppy?
                        JR      NZ,SETSYS1      ;Go if not
                        LD      A,(IY+4)        ;Get data
                        AND     50H             ;Bit 6/4
                        CP      40H             ;DD not alien?
                        JR      NZ,SETSYS1      ;Go if not
                        LD      HL,HITBUF       ;Init buffer
                        LD      D,(IY+9)        ;Get dir cyl
                        LD      E,0             ;Init GAT table
311B                    @@RDSEC                 ;Read GAT table
                        LD      A,49
                        RST     40
                        CP      6               ;Directory read?
                        JP      NZ,IOERR        ;Go on disk error
                        LD      A,(HITBUF+0CDH) ;Get data byte
                        BIT     7,A             ;System disk?
                SETSYS1 LD      DE,0<8+2        ;Init cyl 0
                        JR      NZ,$+3          ;Go if not system
                        INC     D               ;Else on cyl 1
                        LD      C,0             ;Drive 0
                        ENDIF
                ;
311B 210030             LD      HL,HITBUF       ;Set disk buffer
```

Format Init Code

```
311E          Ø443Ø          @@RDSEC                     ;Read sysinfo sector
311E 3E31     ØØØ94          LD      A,49
312Ø EF       ØØØ95          RST     4Ø
3121 C2A529   Ø444Ø          JP      NZ,IOERR            ;Quit on read error
3124 2E76     Ø445Ø          LD      L,7ØH+6             ;Pt to default DCTs
              Ø446Ø ;
              Ø447Ø ;        Establish the default BOOT step rate
              Ø448Ø ;
3126 E5       Ø449Ø          PUSH    HL                  ;Pt IY to the
3127 FDE1     Ø45ØØ          POP     IY                  ;  start of the DCTs
3129 FD7EFD   Ø451Ø          LD      A,(IY+3-6)          ;P/u DCT$ default step
312C E6Ø3     Ø452Ø          AND     3                   ;  & strip off
312E 329731   Ø453Ø          LD      (STEPARM+1),A       ;Keep for Step parm
              Ø454Ø ;
              Ø455Ø ;        Keep cyl count on all 8 drives
              Ø456Ø ;
3131 Ø6Ø8     Ø457Ø          LD      B,8
3133 DD212136 Ø458Ø          LD      IX,DCTCYL           ;Pt to where to stuff
3137 11ØAØØ   Ø459Ø          LD      DE,1Ø               ;  1Ø-byte increments
313A 7E       Ø46ØØ DCTLP1   LD      A,(HL)              ;P/u default # CYL
313B DD77ØØ   Ø461Ø          LD      (IX),A              ;Save in table
313E DD23     Ø462Ø          INC     IX
314Ø 19       Ø463Ø          ADD     HL,DE
3141 1ØF7     Ø464Ø          DJNZ    DCTLP1              ;Loop for 8 DCTs
              Ø465Ø ;
3143 E1       Ø466Ø          POP     HL                  ;Rcvr ptr to cmdline
3144 7E       Ø467Ø FMT1     LD      A,(HL)              ;Ignore spaces
3145 23       Ø468Ø          INC     HL
3146 FE2Ø     Ø469Ø          CP      ' '
3148 28FA     Ø47ØØ          JR      Z,FMT1
314A FE3A     Ø471Ø          CP      ':'                 ;Colon drive indicator?
314C 281F     Ø472Ø          JR      Z,FMT2              ;Go on drive entry
              Ø473Ø ;
              Ø474Ø ;        Drive not entered, prompt for it
              Ø475Ø ;
314E 2B       Ø476Ø          DEC     HL                  ;Backspace command line
314F 2B       Ø477Ø          DEC     HL                  ;  & adjust for next INC
315Ø E5       Ø478Ø          PUSH    HL                  ;Save pointer
3151          Ø479Ø WHDRV    @@DSPLY WHDRV$              ;"which drive...
              ØØØ96          IFEQ    Ø1H,1
3151 218937   ØØØ97          LD      HL,WHDRV$
              ØØØ98          ENDIF
3154 3EØA     ØØØ99          LD      A,1Ø
3156 EF       ØØ1ØØ          RST     4Ø
3157 21ØØ3Ø   Ø48ØØ          LD      HL,HITBUF           ;Input buffer for now
315A Ø1ØØØ1   Ø481Ø          LD      BC,1<8              ;Max 1 char
315D          Ø482Ø          @@KEYIN                     ;Get a 1-char line
315D 3EØ9     ØØ1Ø1          LD      A,9
315F EF       ØØ1Ø2          RST     4Ø
316Ø DAB929   Ø483Ø          JP      C,FMTABT            ;Quit on Break
3163 7E       Ø484Ø          LD      A,(HL)              ;P/u the entry
3164 D63Ø     Ø485Ø          SUB     'Ø'                 ;Cvrt to binary
3166 FEØ8     Ø486Ø          CP      8                   ;Error if > 7
3168 3ØE7     Ø487Ø          JR      NC,WHDRV
316A E1       Ø488Ø          POP     HL                  ;Rcvr command pointer
316B 18Ø8     Ø489Ø          JR      FMT2A
              Ø49ØØ ;
              Ø491Ø ;        Drive entered
              Ø492Ø ;
```

Format Init Code

```
316D 7E         Ø493Ø FMT2     LD      A,(HL)          ;P/u drive #
316E D63Ø       Ø494Ø          SUB     'Ø'             ;Cvrt to ASCII
317Ø FEØ8       Ø495Ø          CP      8               ;Make sure not > 7
3172 D24F36     Ø496Ø          JP      NC,PRMERR
3175 322B2A     Ø497Ø FMT2A    LD      (FMTDRV+1),A    ;Stuff drive
3178 23         Ø498Ø          INC     HL              ;Bump cmdline ptr
3179 116Ø36     Ø499Ø          LD      DE,PRMTBL$      ;Parse any parameters
317C            Ø5ØØØ          @@PARAM
317C 3E11       ØØ1Ø3          LD      A,17
317E EF         ØØ1Ø4          RST     4Ø
317F C24F36     Ø501Ø          JP      NZ,PRMERR       ;Jump on parm error
                Ø502Ø ;
                Ø503Ø ;        Test if any other parm was entered
                Ø504Ø ;
3182 11ØØØØ     Ø505Ø SDPARM   LD      DE,Ø            ;Single density parm
3185 7A         Ø506Ø          LD      A,D
3186 B3         Ø507Ø          OR      E               ;Merge all theses parms
3187 11ØØØØ     Ø508Ø DDPARM   LD      DE,Ø            ;Double density parm
318A B2         Ø509Ø          OR      D
318B B3         Ø510Ø          OR      E
318C 11ØØØØ     Ø511Ø SIDES    LD      DE,Ø            ;Sides parm
318F B2         Ø512Ø          OR      D
319Ø B3         Ø513Ø          OR      E
3191 11ØØØØ     Ø514Ø CPARM    LD      DE,Ø            ;Cylinder parm
3194 B2         Ø515Ø          OR      D
3195 B3         Ø516Ø          OR      E
3196 11ØØFF     Ø517Ø STEPARM  LD      DE,ØFFØØH       ;Init to show if entry
3199 14         Ø518Ø          INC     D               ;Did user enter it?
319A B2         Ø519Ø          OR      D               ;Ø=no user entry
319B 32AC32     Ø520Ø          LD      (PRMMRG+1),A    ;Set to non-zero if any
                Ø521Ø ;
                Ø522Ø ;        If Q-parm, then set NAME & MPW if not entered
                Ø523Ø ;
319E ED5BB132   Ø524Ø          LD      DE,(QPARM+1)    ;P/u Query parm
31A2 2AEB31     Ø525Ø          LD      HL,(NPARM+1)    ;P/u Name parm
31A5 7C         Ø526Ø          LD      A,H
31A6 B5         Ø527Ø          OR      L
31A7 2ØØ4       Ø528Ø          JR      NZ,$+6          ;Go if user entered name
31A9 ED53EB31   Ø529Ø          LD      (NPARM+1),DE    ;  else use Q-parm value
31AD 2A3B32     Ø530Ø          LD      HL,(MPARM+1)    ;P/u Password parm
31BØ 7C         Ø531Ø          LD      A,H
31B1 B5         Ø532Ø          OR      L
31B2 2ØØ4       Ø533Ø          JR      NZ,$+6          ;Go if user entered password
31B4 ED533B32   Ø534Ø          LD      (MPARM+1),DE    ;Set to Q-parm entry
                Ø535Ø ;
31B8 3A2B2A     Ø536Ø          LD      A,(FMTDRV+1)    ;P/u drive
31BB 4F         Ø537Ø          LD      C,A             ;Set in drive register
31BC 212136     Ø538Ø          LD      HL,DCTCYL       ;Find default # cyls
31BF 85         Ø539Ø          ADD     A,L             ;Index the DCTCYL table
31CØ 6F         Ø540Ø          LD      L,A             ;  according to drive #
31C1 8C         Ø541Ø          ADC     A,H
31C2 95         Ø542Ø          SUB     L
31C3 67         Ø543Ø          LD      H,A
31C4 7E         Ø544Ø          LD      A,(HL)          ;P/u cylinder count
31C5 3C         Ø545Ø          INC     A               ;Offset from 1
31C6 323D33     Ø546Ø          LD      (PCYL2+1),A     ;Stuff default for 5"
31C9            Ø547Ø          @@GTDCT                 ;Find the DCT pointer
31C9 3E51       ØØ1Ø5          LD      A,81
31CB EF         ØØ1Ø6          RST     4Ø
```

Format Init Code

```
31CC FDE5       Ø548Ø          PUSH     IY
31CE E1         Ø549Ø          POP      HL              ;Xfer DCT to HL
31CF 11DD2A     Ø55ØØ          LD       DE,SYSDCT       ;Save the system's DCT
31D2 Ø1ØAØØ     Ø551Ø          LD       BC,1Ø           ;  for the drive since
31D5 EDBØ       Ø552Ø          LDIR                     ;  we are altering it
31D7 3A1C26     Ø553Ø          LD       A,(SYSPRM+1)    ;Check if "SYSTEM" parm
31DA 3C         Ø554Ø          INC      A               ; entered
31DB 2ØØ7       Ø555Ø          JR       NZ,FMT2B        ;Go if not
31DD FDCBØ35E   Ø556Ø          BIT      3,(IY+3)        ;Check if hard drive
31E1 CA4936     Ø557Ø          JP       Z,NOTHARD       ;Can't "SYSTEM" floppy
31E4 CDF629     Ø558Ø FMT2B    CALL     DRVNOP          ;Test if drive enabled
31E7 C2A529     Ø559Ø          JP       NZ,IOERR
31EA 21ØØØØ     Ø56ØØ NPARM    LD       HL,Ø            ;NAME parm entered?
31ED 7C         Ø561Ø          LD       A,H
31EE B5         Ø562Ø          OR       L
31EF 3C         Ø563Ø          INC      A               ;Was it just NAME?
31FØ 2826       Ø564Ø          JR       Z,DSKNAM        ;Prompt if so
31F2 3D         Ø565Ø          DEC      A               ;If entered, use it
31F3 2ØØ3       Ø566Ø          JR       NZ,$+5
31F5 21Ø63A     Ø567Ø DFTNAM   LD       HL,PAKNAM$
31F8 11DØ2E     Ø568Ø          LD       DE,GATBUF+ØDØH  ;Yes, move name to field
31FB Ø6Ø8       Ø569Ø          LD       B,8             ;8-chars max
31FD 7E         Ø57ØØ MOVNAM   LD       A,(HL)          ;P/u a char
31FE FE22       Ø571Ø          CP       '"'             ;Closing "
32ØØ 2829       Ø572Ø          JR       Z,CKNAME        ;Exit if end of parm
32Ø2 FE2Ø       Ø573Ø          CP       2ØH             ;Permit all but controls
32Ø4 DA2B32     Ø574Ø          JP       C,CKNAME
32Ø7 FE61       Ø575Ø          CP       'a'             ;If char is lower case,
32Ø9 38Ø6       Ø576Ø          JR       C,MOVNAM1
32ØB FE7B       Ø577Ø          CP       'z'+1
32ØD 3ØØ2       Ø578Ø          JR       NC,MOVNAM1
32ØF EE2Ø       Ø579Ø          XOR      2ØH             ;  make it UC
3211 12         Ø58ØØ MOVNAM1  LD       (DE),A          ;Put char in buffer
3212 23         Ø581Ø          INC      HL              ;Bump both ptrs
3213 13         Ø582Ø          INC      DE
3214 1ØE7       Ø583Ø          DJNZ     MOVNAM          ;Loop til complete
3216 1813       Ø584Ø          JR       CKNAME          ;Check if valid name
                Ø585Ø ;
                Ø586Ø ;        Prompt user for name parameter
                Ø587Ø ;
3218            Ø588Ø DSKNAM   @@DSPLY  DSKNAM$         ;"diskette name?
                ØØ1Ø7          IFEQ     Ø1H,1
3218 21A637     ØØ1Ø8          LD       HL,DSKNAM$
                ØØ1Ø9          ENDIF
321B 3EØA       ØØ11Ø          LD       A,1Ø
321D EF         ØØ111          RST      4Ø
321E CD652A     Ø589Ø          CALL     GET8            ;Get 8 chars, make UC
3221 28D2       Ø59ØØ          JR       Z,DFTNAM        ;Use default if no entry
3223 48         Ø591Ø          LD       C,B             ;Only move to name field
3224 Ø6ØØ       Ø592Ø          LD       B,Ø             ;  how many were entered
3226 11DØ2E     Ø593Ø          LD       DE,GATBUF+ØDØH
3229 EDBØ       Ø594Ø          LDIR
322B 11DØ2E     Ø595Ø CKNAME   LD       DE,GATBUF+ØDØH  ;Now check if illegal
322E CDBC35     Ø596Ø          CALL     CKMPWØ          ;  chars in name
3231 C24136     Ø597Ø          JP       NZ,BADNAM       ;  & quit if so
3234 21D82E     Ø598Ø GETDAT   LD       HL,GATBUF+ØD8H  ;Get today's date & stuff
3237            Ø599Ø          @@DATE
3237 3E12       ØØ112          LD       A,18
3239 EF         ØØ113          RST      4Ø
```

Format Init Code

```
                ;
                ;       Master Password handling
                ;
323A 21ØØØØ     MPARM   LD      HL,Ø            ;Did user enter the MPW?
323D 7C                 LD      A,H
323E B5                 OR      L
323F 3C                 INC     A               ;If only MPW, then prompt
324Ø 2821               JR      Z,MPW           ;Go prompt if not
3242 3D                 DEC     A
3243 2ØØ3               JR      NZ,$+5          ;If entered, use it
3245 21ØE3A     DFTMPW  LD      HL,PAKMPW$      ; else use ours
3248 115736             LD      DE,MPWBUF       ;Shift to pswd field
324B Ø6Ø8               LD      B,8
324D 7E         MOVMPW  LD      A,(HL)
324E FE3Ø               CP      3ØH             ;No spaces permitted
325Ø 3819               JR      C,PRSMPW        ;End also on closing "
3252 FE61               CP      'a'             ;Need cvrt to UC?
3254 38Ø6               JR      C,MOVMPW1
3256 FE7B               CP      'z'+1
3258 3ØØ2               JR      NC,MOVMPW1
325A EE2Ø               XOR     2ØH             ;Cvrt to UC
325C 12         MOVMPW1 LD      (DE),A          ;Store the char and
325D 13                 INC     DE              ; bump the buffer ptrs
325E 23                 INC     HL
325F 1ØEC               DJNZ    MOVMPW
3261 18Ø8               JR      PRSMPW          ;Check if valid password
                ;
                ;       Prompt for master password
                ;
3263 21B737     MPW     LD      HL,MPW$         ;"master...
3266 CD9535             CALL    INPMPW
3269 3ØDA               JR      NC,DFTMPW       ;Use default on <ENTER>
                ;
                ;       Parse the password & stuff into GAT sector buffer
                ;
326B 115736     PRSMPW  LD      DE,MPWBUF
326E CDB535             CALL    CKMPW           ;Check for valid MPW
3271 C2A529             JP      NZ,IOERR
3274 22CE2E             LD      (GATBUF+ØCEH),HL        ;Stuff it
3277 FDCBØ466           BIT     4,(IY+4)        ;Jump if alien controller
327B C2E133             JP      NZ,CALCGPC
327E 212936             LD      HL,TBLDATA      ;Pt to config tables
3281 11Ø6ØØ             LD      DE,6            ;Index the table
3284 FDCBØ36E           BIT     5,(IY+3)        ;8" drive?
3288 28Ø2               JR      Z,INITDEN       ;Bypass if not
328A 19                 ADD     HL,DE           ; else move to 8" configs
328B 19                 ADD     HL,DE
328C 22EF32     INITDEN LD      (SETSDEN+1),HL  ; & stuff for SDEN option
328F EB                 EX      DE,HL           ;6->HL, SDEN->DE
329Ø 19                 ADD     HL,DE           ;Pt to DDEN index table
3291 22DF32             LD      (SETDDEN+1),HL  ;Stuff DDEN config ptr
3294 EB                 EX      DE,HL           ;HL=SDEN, DE=DDEN
3295 FDCBØ3B6           RES     6,(IY+3)        ;Set DCT to SDEN
3299 FDCBØ476           BIT     6,(IY+4)        ;Test if DDEN capability
329D 28Ø5               JR      Z,SETSTD        ;Go if single
329F EB                 EX      DE,HL           ;HL->DDEN table
32AØ FDCBØ3F6           SET     6,(IY+3)        ;Set DCT to DDEN
32A4 CD5935     SETSTD  CALL    SETUP           ;Init to std config
32A7 FDCBØ4AE           RES     5,(IY+4)        ;Set to 1-sided
```

Format Init Code

```
32AB 3EØØ      Ø659Ø PRMMRG  LD      A,Ø             ;<>Ø if config parms
32AD B7        Ø66ØØ         OR      A               ;  in command line
32AE 2ØØ8      Ø661Ø         JR      NZ,GETDEN
32BØ 11FFFF    Ø662Ø QPARM   LD      DE,-1           ;Prompts? Default=Y
32B3 7A        Ø663Ø         LD      A,D
32B4 B3        Ø664Ø         OR      E
32B5 CAD633    Ø665Ø         JP      Z,PSTEP1        ;Go if no prompting
32B8 FDCBØ476  Ø666Ø GETDEN  BIT     6,(IY+4)        ;Bypass DDEN request msg
32BC 283A      Ø667Ø         JR      Z,PMTSIDE       ;  if no DDEN capability
32BE 3AAC32    Ø668Ø         LD      A,(PRMMRG+1)    ;Also, don't prompt if
32C1 B7        Ø669Ø         OR      A               ;  any config parm was
32C2 2Ø13      Ø67ØØ         JR      NZ,GDDEN1       ;  entered with command
32C4 216538    Ø671Ø         LD      HL,DEN?$        ;Density <S,D>...
32C7 CD5D2A    Ø672Ø         CALL    GET3
32CA 282C      Ø673Ø         JR      Z,PMTSIDE       ;Go on <ENTER>
32CC 7E        Ø674Ø         LD      A,(HL)          ;P/u respsonse
32CD FE53      Ø675Ø         CP      'S'             ;Single Density?
32CF 281D      Ø676Ø         JR      Z,SETSDEN
32D1 FE44      Ø677Ø         CP      'D'             ;Double density?
32D3 28Ø9      Ø678Ø         JR      Z,SETDDEN
32D5 18E1      Ø679Ø         JR      GETDEN          ;Redo if bad response
32D7 3A8831    Ø68ØØ GDDEN1  LD      A,(DDPARM+1)    ;Not prompted, was DDEN
32DA EEFF      Ø681Ø         XOR     -1              ;  set in command line?
32DC 2ØØ9      Ø682Ø         JR      NZ,GSDEN1       ;Bypass if not
32DE 21ØØØØ    Ø683Ø SETDDEN LD      HL,$-$          ;P/u DDEN index table
32E1 FDCBØ3F6  Ø684Ø         SET     6,(IY+3)        ;Set DCT to DDEN
32E5 18ØE      Ø685Ø         JR      CHGDEN
32E7 3A8331    Ø686Ø GSDEN1  LD      A,(SDPARM+1)    ;Was SDEN parm
32EA EEFF      Ø687Ø         XOR     -1              ;  on command line?
32EC 2ØØA      Ø688Ø         JR      NZ,PMTSIDE      ;Go if not
32EE 21ØØØØ    Ø689Ø SETSDEN LD      HL,$-$          ;P/u SDEN index table
32F1 FDCBØ3B6  Ø69ØØ         RES     6,(IY+3)        ;Set DCT to SDEN
32F5 CD5935    Ø691Ø CHGDEN  CALL    SETUP           ;Init #CYLs & alloc
32F8 3AAC32    Ø692Ø PMTSIDE LD      A,(PRMMRG+1)    ;Config parms entered
32FB B7        Ø693Ø         OR      A               ;On command line?
32FC 2Ø2Ø      Ø694Ø         JR      NZ,PMTS1        ;Bypass if yes
32FE FDE5      Ø695Ø         PUSH    IY              ;P/u flag table
33ØØ           Ø696Ø         @@FLAGS                 ;  and check if
33ØØ 3E65      ØØ114         LD      A,1Ø1
33Ø2 EF        ØØ115         RST     4Ø
33Ø3 FDCBØB6E  Ø697Ø         BIT     5,(IY+'L'-'A')  ;  2-side inhibit?
33Ø7 FDE1      Ø698Ø         POP     IY
33Ø9 2Ø13      Ø699Ø         JR      NZ,PMTS1        ;If set, use 1 side
33ØB 214638    Ø7ØØØ         LD      HL,SIDES$       ;"double sided...?
33ØE CD5D2A    Ø7Ø1Ø         CALL    GET3            ;Get # sides wanted
3311 2816      Ø7Ø2Ø         JR      Z,PMTCYL        ;Go on <ENTER>
3313 7E        Ø7Ø3Ø         LD      A,(HL)          ;P/u response char
3314 FE31      Ø7Ø4Ø         CP      '1'             ;1 is ok
3316 2811      Ø7Ø5Ø         JR      Z,PMTCYL
3318 FE32      Ø7Ø6Ø         CP      '2'             ;  and so is 2
331A 2ØDC      Ø7Ø7Ø         JR      NZ,PMTSIDE      ;  but redo on anything else
331C 18Ø5      Ø7Ø8Ø         JR      TSTSID
               Ø7Ø9Ø ;
               Ø71ØØ ;       Check side parm from command line
               Ø711Ø ;
331E 3A8D31    Ø712Ø PMTS1   LD      A,(SIDES+1)     ;How many sides?
3321 FEØ2      Ø713Ø         CP      2
3323 2ØØ4      Ø714Ø TSTSID  JR      NZ,PMTCYL       ;DCT ok if not 2
3325 FDCBØ4EE  Ø715Ø         SET     5,(IY+4)        ;Set 2-sided drive
```

Format Init Code

```
3329 FD7EØ3     PMTCYL  LD      A,(IY+3)        ;No cylinder request
332C E628               AND     28H             ;  if either hard drive
332E 2Ø33               JR      NZ,PMTSTEP      ;  or 8" drive
333Ø 3AAC32     PCYL1   LD      A,(PRMMRG+1)    ;P/u config test byte &
3333 B7                 OR      A               ;  bypass cyl req if user
3334 2Ø19               JR      NZ,PCYL4        ;  entered cmd line parms
3336 21CA37             LD      HL,NUMCYL$      ;"number of cyls..?
3339 CD5D2A             CALL    GET3
333C 3EØØ       PCYL2   LD      A,Ø             ;P/u default # cyls
333E C48235             CALL    NZ,CVBIN        ;Get # of cyls on CR
3341 FE61       PCYL3   CP      96+1            ;System cannot support
3343 3ØEB               JR      NC,PCYL1        ;  anything over 96 (95)
3345 FE23               CP      35
3347 38E7               JR      C,PCYL1         ;Must be 35 or more
3349 3D                 DEC     A               ;Adjust to zero offset
334A FD77Ø6             LD      (IY+6),A        ;  & stuff in DCT
334D 1814               JR      PMTSTEP
                ;
                ;       User entered config parms with command line
                ;
334F 3A9231     PCYL4   LD      A,(CPARM+1)     ;Was cyl= one of them?
3352 B7                 OR      A
3353 28ØE               JR      Z,PMTSTEP       ;Bypass if not
3355 FE61               CP      96+1
3357 D24F36             JP      NC,PRMERR       ;Parm error if too big
335A FE23               CP      35
335C DA4F36             JP      C,PRMERR        ;  or too small
335F 3D                 DEC     A               ;Adjust to zero offset
336Ø FD77Ø6             LD      (IY+6),A        ;  & stuff into DCT
3363 FDCBØ466   PMTSTEP BIT     4,(IY+4)        ;Alien controller?
3367 2Ø8F               JR      NZ,PMTSIDE      ;No adjustable step rate if so
                ;
                ;       If step rate parm wasn't entered, prompt
                ;       for it but first determine 8" or 5" drive
                ;
3369 3AAC32             LD      A,(PRMMRG+1)    ;Did user enter config
336C B7                 OR      A               ;Parms on command line?
336D 2Ø67               JR      NZ,PSTEP1       ;Go to step prompt if yes
                ;
336F FDE5               PUSH    IY              ;P/u flag table and
3371                    @@FLAGS                 ;  check if
3371 3E65               LD      A,1Ø1
3373 EF                 RST     4Ø
3374 FDCBØB46           BIT     Ø,(IY+'L'-'A')  ;  step prompt inhibited
3378 FDE1               POP     IY
337A 2Ø5A               JR      NZ,PSTEP1       ;Bypass if set
                ;
337C FDCBØ36E           BIT     5,(IY+3)        ;Need prompt, 8"?
338Ø 2Ø2A               JR      NZ,STEP8        ;Jump if 8"
                ;
                ;       5" drive step rate parsing
                ;
3382 21E137     STEP5   LD      HL,STEP5$       ;"...step rate - 5"
3385 CD5D2A             CALL    GET3
3388 CD8235             CALL    CVBIN           ;Get 5" step rate
338B B7                 OR      A               ;Use default?
338C 2848               JR      Z,PSTEP1        ;Go if parm not entered
338E Ø6ØØ               LD      B,Ø             ;Init key to Ø
339Ø FEØ6               CP      6
```

Format Init Code

```
3392 2849      Ø773Ø          JR      Z,GOTSTEP
3394 Ø6Ø1      Ø774Ø          LD      B,1             ;Init key to 1
3396 FEØC      Ø775Ø          CP      12
3398 2843      Ø776Ø          JR      Z,GOTSTEP
339A Ø6Ø2      Ø777Ø          LD      B,2             ;Init key to 2
339C FE14      Ø778Ø          CP      2Ø
339E 283D      Ø779Ø          JR      Z,GOTSTEP
33AØ Ø6Ø3      Ø78ØØ          LD      B,3             ;Init key to 3
33A2 FE1E      Ø781Ø          CP      3Ø
33A4 2837      Ø782Ø          JR      Z,GOTSTEP
33A6 FE28      Ø783Ø          CP      4Ø
33A8 2833      Ø784Ø          JR      Z,GOTSTEP
33AA 18D6      Ø785Ø          JR      STEP5           ;Re-request, bad value
               Ø786Ø ;
               Ø787Ø ;        8" drive step rate parsing
               Ø788Ø ;
33AC 211338    Ø789Ø STEP8    LD      HL,STEP8$       ;"step rate - 8"...
33AF CD5D2A    Ø79ØØ          CALL    GET3
33B2 CD8235    Ø791Ø          CALL    CVBIN           ;Get 8" step rate
33B5 B7        Ø792Ø          OR      A               ;Use default?
33B6 281E      Ø793Ø          JR      Z,PSTEP1        ;Go if not entered
33B8 Ø6ØØ      Ø794Ø          LD      B,Ø             ;Init key to Ø
33BA FEØ3      Ø795Ø          CP      3
33BC 281F      Ø796Ø          JR      Z,GOTSTEP
33BE Ø6Ø1      Ø797Ø          LD      B,1             ;Init key to 1
33CØ FEØ6      Ø798Ø          CP      6
33C2 2819      Ø799Ø          JR      Z,GOTSTEP
33C4 Ø6Ø2      Ø8ØØØ          LD      B,2             ;Init key to 2
33C6 FEØA      Ø8Ø1Ø          CP      1Ø
33C8 2813      Ø8Ø2Ø          JR      Z,GOTSTEP
33CA Ø6Ø3      Ø8Ø3Ø          LD      B,3             ;Init key to 3
33CC FEØF      Ø8Ø4Ø          CP      15
33CE 28ØD      Ø8Ø5Ø          JR      Z,GOTSTEP
33DØ FE14      Ø8Ø6Ø          CP      2Ø
33D2 28Ø9      Ø8Ø7Ø          JR      Z,GOTSTEP
33D4 18D6      Ø8Ø8Ø          JR      STEP8           ;Bad entry, re-request
33D6 3A9731    Ø8Ø9Ø PSTEP1   LD      A,(STEPARM+1)   ;P/u step parm entry
33D9 E6Ø3      Ø81ØØ          AND     3               ;Keep 2 lo-order bits
33DB 18Ø1      Ø811Ø          JR      $+3
33DD 78        Ø812Ø GOTSTEP  LD      A,B             ;Stuff boot step rate key
33DE 32E72A    Ø813Ø          LD      (STEPDFT),A
               Ø814Ø ;
               Ø815Ø ;        Routine to calculate the # of grans per logical
               Ø816Ø ;        cylinder so that the GAT byte can be constructed
               Ø817Ø ;
33E1 FD7EØ8    Ø818Ø CALCGPC  LD      A,(IY+8)        ;P/u # of grans per cyl
33E4 Ø7        Ø819Ø          RLCA                    ;Rotate to bits Ø-2
33E5 Ø7        Ø82ØØ          RLCA
33E6 Ø7        Ø821Ø          RLCA
33E7 E6Ø7      Ø822Ø          AND     7               ;Strip off other data
33E9 3C        Ø823Ø          INC     A               ;Adj for zero offset
               Ø824Ø ;
               Ø825Ø ;        If double siding (cylindering), double the count
               Ø826Ø ;
33EA FDCBØ46E  Ø827Ø          BIT     5,(IY+4)        ;Test if 2-sided drive
33EE 28Ø1      Ø828Ø          JR      Z,$+3           ;Bypass if only 1-sided
33FØ 87        Ø829Ø          ADD     A,A             ;Double the grans/cyl
33F1 Ø1FFFF    Ø83ØØ          LD      BC,ØFFFFH       ;Init GAT byte to ones
33F4 CB2Ø      Ø831Ø CGPC1    SLA     B               ;Now keep removing low
```

Format Init Code

```
33F6 3D        08320          DEC     A                 ;  order bits , 1 bit for
33F7 20FB      08330          JR      NZ,CGPC1          ;  each available granule
33F9 21002E    08340          LD      HL,GATBUF         ;Pt to GAT buffer area
33FC FD7E06    08350          LD      A,(IY+6)          ;P/u highest # cylinder
33FF 70        08360 CGPC2    LD      (HL),B            ;Stuff the GAT byte into
3400 2C        08370          INC     L                 ;Each position of the GAT
3401 BD        08380          CP      L                 ;One byte per cylinder
3402 30FB      08390          JR      NC,CGPC2
               08400 ;
               08410 ;        Test if we are at 202 first by ignoring the
               08420 ;        first two instructions with LD DE,xxxx
               08430 ;
3404 3ECB      08440          LD      A,0CBH            ;Continue to stuff GAT
3406 11        08450          DB      11H               ;  until cyl 202
3407 71        08460 CGPC3    LD      (HL),C            ;Use FFH to show unused
3408 2C        08470          INC     L
3409 BD        08480          CP      L                 ;First test here for
340A 20FB      08490          JR      NZ,CGPC3          ;  match against 202
               08500 ;
               08510 ;        Prompt for destination disk & prepare it
               08520 ;
340C 3A2B2A    08530          LD      A,(FMTDRV+1)      ;P/u drive
340F B7        08540          OR      A
3410 2020      08550          JR      NZ,PFMT1          ;Bypass if other than 0
3412           08560 PMTDST   @@DSPLY PMTDST$           ;"load dest disk...
               00118          IFEQ    01H,1
3412 21C038    00119          LD      HL,PMTDST$
               00120          ENDIF
3415 3E0A      00121          LD      A,10
3417 EF        00122          RST     40
3418 FDE5      08570          PUSH    IY                ;Save DCT pointer
341A           08580          @@FLAGS                   ;Point to flags
341A 3E65      00123          LD      A,101
341C EF        00124          RST     40
341D FDCB126E  08590          BIT     5,(IY+'S'-'A')    ;Check for JCL active
3421 FDE1      08600          POP     IY                ;Restore pointer
3423 C2B929    08610          JP      NZ,FMTABT         ;Abort if in JCL
3426 210030    08620          LD      HL,HITBUF
3429 010000    08630          LD      BC,0              ;Zero characters means
342C           08640          @@KEYIN                   ;Enter or Break only
342C 3E09      00125          LD      A,9
342E EF        00126          RST     40
342F DAB929    08650          JP      C,FMTABT          ;Abort if Break
3432 FDE5      08660 PFMT1    PUSH    IY                ;Xfer DCT ptr to HL
3434 E1        08670          POP     HL                ;  & move DCT again
3435 111736    08680          LD      DE,TMPDCT         ;  to store tempy
3438 010A00    08690          LD      BC,10
343B EDB0      08700          LDIR
               08710          IF      @MOD2
               08720          CALL    SELECT
               08730          JP      NZ,IOERR          ;Go on error
               08740          ENDIF
343D CDFF29    08750          CALL    RESTOR            ;Restore to cyl 0
3440 C2A529    08760          JP      NZ,IOERR          ;Go on error
3443 CD092A    08770          CALL    RSELCT            ;Reselect drive
3446 C2A529    08780          JP      NZ,IOERR          ;Go on error
3449 FDCB0466  08790          BIT     4,(IY+4)          ;Jump if alien controller
344D 2040      08800          JR      NZ,PFMT3
344F 218738    08810          LD      HL,NOTRDY$        ;Init "drive not ready
```

Format Init Code

```
3452 CB7F                BIT     7,A             ;Test FDC status for READY
3454 C2BC29              JP      NZ,EXTERR       ;Quit if not ready
3457 21AC38              LD      HL,NODRV$       ;Init "drive not in...
345A CB57                BIT     2,A             ;Test FDC status for TRACK-Ø
345C CABC29              JP      Z,EXTERR        ;  & error if not at track Ø
345F CDEC35              CALL    CKDRV           ;Ck if floppy not present
3462 2ØAE                JR      NZ,PMTDST
3464 219738              LD      HL,CANTWR$      ;Init "write protected..
3467 Ø7                  RLCA                    ;Align to bit 7
3468 FDB6Ø3              OR      (IY+3)          ;Combine with soft WP
346B E68Ø                AND     8ØH             ;WP error?
346D C2BC29              JP      NZ,EXTERR       ;Can't format over WP
347Ø 3A1C26              LD      A,(SYSPRM+1)    ;Don't check space needed
3473 B7                  OR      A               ;  if SYSTEM info only
3474 2Ø19                JR      NZ,PFMT3
3476 21ØØ31              LD      HL,FORMAT       ;Start of format buffer
3479 11ØØØØ      PFMT2   LD      DE,Ø            ;P/u format space needed
347C 19                  ADD     HL,DE           ;Pt to last addr needed
347D 54                  LD      D,H             ;Xfer to reg DE
347E 5D                  LD      E,L
347F 21ØØØØ              LD      HL,Ø            ;Set up for HIGH$ fetch
3482 45                  LD      B,L
3483                     @@HIGH$                 ;Make sure it won't wrap
3483 3E64                LD      A,1ØØ
3485 EF                  RST     4Ø
3486 AF                  XOR     A
3487 ED52                SBC     HL,DE           ;  into protected memory
3489 216Ø37              LD      HL,NOMEM$       ;Init "insufficient mem..
348C DABC29              JP      C,EXTERR        ;Quit if no memory available
348F 11ØØØØ      PFMT3   LD      DE,Ø            ;Init to cyl Ø, sect Ø
3492 CD272A              CALL    VERSEC          ;Verify BOOT
3495 C24535              JP      NZ,PFMT6        ;Assume unformatted if err
                 ;
                 ;       Appears formatted, is there SYSTEM information?
                 ;
3498 3A1C26              LD      A,(SYSPRM+1)    ;Ignore data if SYSTEM
349B B7                  OR      A               ;  info only
349C C24535              JP      NZ,PFMT6
349F 21ØØ3Ø              LD      HL,HITBUF       ;Pt to i/o buffer
34A2 CD222A              CALL    RDSEC           ;Now try to read BOOT
34A5 C2A529              JP      NZ,IOERR        ;Jump on error
34A8                     @@LOGOT HASDAT$         ;Show "disk contains data
                         IFEQ    Ø1H,1
34A8 21E338              LD      HL,HASDAT$
                         ENDIF
34AB 3EØC                LD      A,12
34AD EF                  RST     4Ø
34AE 21FA38              LD      HL,NOFMT$       ;Init "non-std format
                 ;
                 ;       BOOT was read, is there a valid directory pointer
                 ;
34B1 3AØ23Ø              LD      A,(HITBUF+2)    ;P/u dir cyl # (possible)
34B4 FDBEØ6              CP      (IY+6)          ;Check against max cyl #
34B7 3Ø69                JR      NC,PFMT5        ;Go if bigger (or =)
                 ;
                 ;       Read the assumed GAT & test it
                 ;
34B9 21ØØ3Ø              LD      HL,HITBUF
34BC 5D                  LD      E,L
```

Format Init Code

```
34BD 57         Ø934Ø          LD      D,A              ;Pt to assumed GAT sector
34BE 21ØØ3Ø     Ø935Ø          LD      HL,HITBUF        ;Pt to buffer
34C1 CD222A     Ø936Ø          CALL    RDSEC            ;Read the sector
34C4 FEØ6       Ø937Ø          CP      6                ;Dir errcod returned?
34C6 28Ø5       Ø938Ø          JR      Z,PFMT4          ;Jump if yes & grab data
34C8 21ØE39     Ø939Ø          LD      HL,CANTRD$       ;Init "unreadable dir...
34CB 1855       Ø94ØØ          JR      PFMT5
34CD 212339     Ø941Ø PFMT4    LD      HL,NODIR$        ;Init "non-init dir
34DØ 3ADA3Ø     Ø942Ø          LD      A,(HITBUF+ØDAH)  ;Check if date field
34D3 FE2F       Ø943Ø          CP      '/'              ;  is present
34D5 2Ø4B       Ø944Ø          JR      NZ,PFMT5         ;Jump if no
                Ø945Ø ;
                Ø946Ø ;        The directory is readable - request its MPW
                Ø947Ø ;
34D7 21DØ3Ø     Ø948Ø          LD      HL,HITBUF+ØDØH
34DA 114239     Ø949Ø          LD      DE,PACKID$+5     ;Move name & date into
34DD Ø1Ø8ØØ     Ø95ØØ          LD      BC,8             ;  display message field
34EØ EDBØ       Ø951Ø          LDIR
34E2 115139     Ø952Ø          LD      DE,PACKID$+14H
34E5 ØEØ8       Ø953Ø          LD      C,8
34E7 EDBØ       Ø954Ø          LDIR
                Ø955Ø ;
                Ø956Ø ;        If MPW = "PASSWORD", just ck ABS
                Ø957Ø ;
34E9 2ACE3Ø     Ø958Ø          LD      HL,(HITBUF+ØCEH)        ;P/u disk MPW
34EC 11EØ42     Ø959Ø          LD      DE,PASSWORD      ;Password=PASSWORD
34EF AF         Ø96ØØ          XOR     A
34FØ ED52       Ø961Ø          SBC     HL,DE            ;Is it password?
34F2 213D39     Ø962Ø          LD      HL,PACKID$       ;Init"Name=, Date=
34F5 282B       Ø963Ø          JR      Z,PFMT5          ;If match, go check ABS
34F7            Ø964Ø          @@LOGOT                  ;Log the ID field
                ØØ134          IFEQ    ØØH,1
                ØØ135          LD      HL,
                ØØ136          ENDIF
34F7 3EØC       ØØ137          LD      A,12
34F9 EF         ØØ138          RST     4Ø
34FA FDE5       Ø965Ø          PUSH    IY               ;Abort if in JCL
34FC            Ø966Ø          @@FLAGS
34FC 3E65       ØØ139          LD      A,1Ø1
34FE EF         ØØ14Ø          RST     4Ø
34FF FDCB126E   Ø967Ø          BIT     5,(IY+'S'-'A')   ;Test if "DOing"
35Ø3 FDE1       Ø968Ø          POP     IY
35Ø5 C2B929     Ø969Ø          JP      NZ,FMTABT        ;Can't get PW if in JCL
                Ø97ØØ ;
                Ø971Ø ;        User must enter Current Pack's MPW to proceed
                Ø972Ø ;
35Ø8 215A39     Ø973Ø OLDMPW   LD      HL,OLDMPW$       ;"What's the old MPW?
35ØB CD9535     Ø974Ø          CALL    INPMPW           ;Grab user input to match
35ØE 3ØF8       Ø975Ø          JR      NC,OLDMPW
351Ø 115736     Ø976Ø          LD      DE,MPWBUF
3513 CDB935     Ø977Ø          CALL    HASHMPW          ;Hash user entry
                Ø978Ø ;
                Ø979Ø ;        Routine to test master password for match
                Ø98ØØ ;
3516 EB         Ø981Ø          EX      DE,HL            ;Xfer hashed MPW to DE
3517 2ACE3Ø     Ø982Ø          LD      HL,(HITBUF+ØCEH)        ;Else grab pack MPW
351A AF         Ø983Ø          XOR     A                ;Clear carry flag
351B ED52       Ø984Ø          SBC     HL,DE            ;Did user enter pack MPW?
351D C24536     Ø985Ø          JP      NZ,BADMPW        ;Abort if no match
```

Format Init Code

```
3520 1823      09860          JR      PFMT6
               09870 ;
               09880 ;        The directory was not readable - req assurance
               09890 ;
3522           09900 PFMT5    @@LOGOT
               00141          IFEQ    00H,1
               00142          LD      HL,
               00143          ENDIF
3522 3E0C      00144          LD      A,12
3524 EF        00145          RST     40
3525 110000    09910 APARM    LD      DE,0             ;ABS parameter
3528 1C        09920          INC     E
3529 281A      09930          JR      Z,PFMT6          ;Go if ABS used
352B FDE5      09940          PUSH    IY
352D           09950          @@FLAGS
352D 3E65      00146          LD      A,101
352F EF        00147          RST     40
3530 FDCB126E  09960          BIT     5,(IY+'S'-'A')   ;Test if "DOing"
3534 FDE1      09970          POP     IY
3536 C2B929    09980          JP      NZ,FMTABT        ;Abort if JCL but no ABS
3539 21B439    09990          LD      HL,SURE?$        ;"are you sure...?
353C CD5D2A    10000          CALL    GET3             ;Get response
353F 7E        10010          LD      A,(HL)
3540 FE59      10020          CP      'Y'              ;If not Yes, abort
3542 C2B929    10030          JP      NZ,FMTABT
3545 FDE5      10040 PFMT6    PUSH    IY               ;Move drive code table
3547 D1        10050          POP     DE               ;  back into place
3548 211736    10060          LD      HL,TMPDCT        ;  into system slot
354B 010A00    10070          LD      BC,10
354E EDB0      10080          LDIR
3550 CDFF29    10090          CALL    RESTOR           ;Restore to cylinder 0
3553 C2A529    10100          JP      NZ,IOERR         ;Go on error
3556 C30126    10110          JP      GOFMT            ;Go and format it
               10120 ;
               10130 ;        Routine to set up the DCT for format
               10140 ;
3559 3A3D33    10150 SETUP    LD      A,(PCYL2+1)      ;P/u the highest # cyl
355C FDCB036E  10160          BIT     5,(IY+3)         ;If 8" drive, use 77
3560 2802      10170          JR      Z,$+4            ;Go if only 5"
3562 3E4D      10180          LD      A,77             ;8" drives are 77 cyls
3564 3D        10190          DEC     A
3565 FD7706    10200          LD      (IY+6),A         ;Stuff in our DCT
3568 5E        10210          LD      E,(HL)           ;Grab address to
3569 23        10220          INC     HL               ;  master formatting table
356A 56        10230          LD      D,(HL)
356B 23        10240          INC     HL
356C ED530926  10250          LD      (FMTTBL+1),DE    ;Stuff for later use
3570 5E        10260          LD      E,(HL)           ;P/u DCT+7 data
3571 23        10270          INC     HL               ;Max sector, # of heads
3572 56        10280          LD      D,(HL)           ;P/u DCT+8 data, # of
3573 23        10290          INC     HL               ;  sectors/gran & grans/cyl
3574 FD7307    10300          LD      (IY+7),E         ;Stuff these values into
3577 FD7208    10310          LD      (IY+8),D         ;  our DCT
357A 5E        10320          LD      E,(HL)           ;P/u space needed for
357B 23        10330          INC     HL               ;  the formatting buffer
357C 56        10340          LD      D,(HL)
357D ED537A34  10350          LD      (PFMT2+1),DE     ;  & stuff that for later
3581 C9        10360          RET
               10370 ;
```

Format Init Code

```
                ;       Convert decimal ASCII to binary
                ;
3582 1EØØ       CVBIN   LD      E,Ø                 ;Init value to Ø
3584 7E         CVB1    LD      A,(HL)              ;Get a character
3585 23                 INC     HL                  ;Bump buff ptr
3586 D63Ø               SUB     3ØH                 ;Make binary
3588 47                 LD      B,A
3589 FEØA               CP      ØAH                 ;Was it a decimal digit?
358B 7B                 LD      A,E
358C DØ                 RET     NC                  ;Return if not
358D 87                 ADD     A,A                 ;Mult previous value X 1Ø
358E 87                 ADD     A,A
358F 83                 ADD     A,E
359Ø 87                 ADD     A,A
3591 8Ø                 ADD     A,B                 ;Add in new digit
3592 5F                 LD      E,A                 ;Put results in E
3593 18EF               JR      CVB1                ;Loop
                ;
3595            INPMPW  @@DSPLY
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
3595 3EØA               LD      A,1Ø
3597 EF                 RST     4Ø
3598 215736             LD      HL,MPWBUF           ;Use this buffer
359B Ø6Ø8               LD      B,8                 ;8 chars max
359D CD6B2A             CALL    GET8A               ;Input the pswd
35AØ C8                 RET     Z                   ;Go if Enter only
35A1 EB                 EX      DE,HL
35A2 83                 ADD     A,E                 ;Find where the X'ØD' was
35A3 6F                 LD      L,A                 ;  stuffed & cover it
35A4 7A                 LD      A,D
35A5 CEØØ               ADC     A,Ø
35A7 67                 LD      H,A
35A8 3EØ8               LD      A,8                 ;If 8 chars entered,
35AA 9Ø                 SUB     B
35AB 37                 SCF                         ;  done
35AC C8                 RET     Z
35AD 47                 LD      B,A                 ;  else pad the buffer
35AE 362Ø       FILLBLK LD      (HL),' '            ;  w/spaces
35BØ 23                 INC     HL
35B1 1ØFB               DJNZ    FILLBLK
35B3 37                 SCF
35B4 C9                 RET
                ;
35B5 CDBC35     CKMPW   CALL    CKMPWØ
35B8 CØ                 RET     NZ
                ;
                ;       Hash a diskette password
                ;
35B9 3EE4       HASHMPW LD      A,ØE4H              ;Use SYS2 routine
35BB EF                 RST     4Ø
                ;
35BC Ø6Ø8       CKMPWØ  LD      B,8                 ;8 char to check
35BE D5                 PUSH    DE                  ;Xfer start of PW
35BF E1                 POP     HL                  ;  to HL
35CØ 7E                 LD      A,(HL)              ;P/u 1st char
35C1 18ØE               JR      CKMPW2              ;  & check <A-Z>
35C3 23         CKMPW1  INC     HL                  ;Advance to next char
```

```
35C4 7E        1Ø92Ø          LD      A,(HL)          ;P/u the char
35C5 FE2Ø      1Ø93Ø          CP      ' '
35C7 2818      1Ø94Ø          JR      Z,CKMPW7        ;Go on space
35C9 FE3Ø      1Ø95Ø          CP      'Ø'
35CB 3818      1Ø96Ø          JR      C,INVMPW        ;Bad if less than o
35CD FE3A      1Ø97Ø          CP      '9'+1           ;  or greater than 9
35CF 38Ø8      1Ø98Ø          JR      C,CKMPW3
35D1 FE41      1Ø99Ø CKMPW2   CP      'A'
35D3 381Ø      11ØØØ          JR      C,INVMPW        ;  but less than A
35D5 FE5B      11Ø1Ø          CP      'Z'+1
35D7 3ØØC      11Ø2Ø          JR      NC,INVMPW       ;More than Z also bad
35D9 1ØE8      11Ø3Ø CKMPW3   DJNZ    CKMPW1          ;Char ok, do another
35DB AF        11Ø4Ø          XOR     A               ;Set Z, PW good
35DC C9        11Ø5Ø          RET
               11Ø6Ø ;
35DD 23        11Ø7Ø CKMPW5   INC     HL              ;Next char position
35DE BE        11Ø8Ø          CP      (HL)            ;No imbedded spaces
35DF 2ØØ4      11Ø9Ø          JR      NZ,INVMPW
35E1 1ØFA      111ØØ CKMPW7   DJNZ    CKMPW5          ;Loop til 8 checked
35E3 AF        1111Ø          XOR     A               ;Set Z = PW good
35E4 C9        1112Ø          RET
               1113Ø ;
35E5 21DA39    1114Ø INVMPW   LD      HL,INVMPW$      ;Init "Invalid PW
35E8 3E3F      1115Ø          LD      A,63            ;Indicate extended error
35EA B7        1116Ø          OR      A               ;Set NZ condition
35EB C9        1117Ø          RET
               1118Ø ;
               1119Ø ;        Brief routine to check a drive for availability
               112ØØ ;
35EC 21ØØ3Ø    1121Ø CKDRV    LD      HL,HITBUF
35EF           1122Ø          @@TIME                  ;P/u the timer pointer
35EF 3E13      ØØ153          LD      A,19
35F1 EF        ØØ154          RST     4Ø
35F2 EB        1123Ø          EX      DE,HL           ;TIME$ to HL
35F3 2B        1124Ø          DEC     HL              ;TIMER$ to HL
35F4 7E        1125Ø          LD      A,(HL)          ;P/u current timer value
35F5 C6ØF      1126Ø          ADD     A,15            ;Set timeout to 5ØØms
35F7 57        1127Ø          LD      D,A             ;Save for test later
               1128Ø ;
               1129Ø ;        Test for diskette in drive & rotating
               113ØØ ;
35F8 CDØ836    1131Ø CKDR1    CALL    CKDR6           ;Test index pulse
35FB 2ØFB      1132Ø          JR      NZ,CKDR1        ;Jump on index
35FD CDØ836    1133Ø CKDR2    CALL    CKDR6           ;Test index pulse
36ØØ 28FB      1134Ø          JR      Z,CKDR2         ;Jump on no index
36Ø2 CDØ836    1135Ø CKDR2A   CALL    CKDR6
36Ø5 2ØFB      1136Ø          JR      NZ,CKDR2A       ;Jump on index
36Ø7 C9        1137Ø          RET
36Ø8 FB        1138Ø CKDR6    EI                      ;Make sure they're ON
36Ø9 7E        1139Ø          LD      A,(HL)          ;P/u latest TIMER$ value
36ØA 92        114ØØ          SUB     D               ;5ØØms passed?
36ØB 28Ø6      1141Ø          JR      Z,CKDR7
36ØD CDØ92A    1142Ø          CALL    RSELCT          ;Select & wait not busy
361Ø CB4F      1143Ø          BIT     1,A             ;Test index
3612 C9        1144Ø          RET
3613 D1        1145Ø CKDR7    POP     DE              ;Pop the ret address
3614 F6Ø1      1146Ø          OR      1               ;Set "Illegal drive #
3616 C9        1147Ø          RET                     ;With NZ
               1148Ø ;
```

Format Init Code

```
                11490 ;         Temporary storage space for format drive DCT
                11500 ;
000A            11510 TMPDCT  DS      10
0008            11520 DCTCYL  DS      8               ;Default # cyls
                11530 ;
                11540 ;         Config table for single density 5"
                11550 ;
3629            11560 TBLDATA EQU     $
3629 EA2A       11570         DW      S5TBL,2409H,3381
     0924 350D
                11580 ;
                11590 ;         Config table for double density 5"
                11600 ;
362F 2A2B       11610         DW      D5TBL,4511H,6506
     1145 6A19
                11620 ;
                11630 ;         Config table for single density 8"
                11640 ;
3635 812B       11650         DW      S8TBL,270FH,5464
     0F27 5815
                11660 ;
                11670 ;         Config table for double density 8"
                11680 ;
363B C62B       11690         DW      D8TBL,491DH,10673
     1D49 B129
                11700 ;
                11710 ;         Parm error exit
                11720 ;
3641 21F439     11730 BADNAM  LD      HL,BADNAM$
3644 DD         11740         DB      0DDH
3645 21DA39     11750 BADMPW  LD      HL,INVMPW$
3648 DD         11760         DB      0DDH
3649 214737     11770 NOTHARD LD      HL,HARD$
364C C3BC29     11780         JP      EXTERR
364F 3E2C       11790 PRMERR  LD      A,44            ;Init Parm ERROR
3651 C3A529     11800         JP      IOERR
                11810 ;
                11820 ;         Load SYS2 overlay
                11830 ;
3654 3E84       11840 GETSYS2 LD      A,84H
3656 EF         11850         RST     28H
                11860 ;
3657 20         11870 MPWBUF  DB      '         '
     20 20 20 20 20 20 20 20
                11880 PRMTBL$
0080            11890 VAL     EQU     80H
0040            11900 SW      EQU     40H
0020            11910 STR     EQU     20H
0010            11920 SGL     EQU     10H
3660 80         11930         DB      80H
3661 74         11940         DB      SW!STR!SGL!4,'NAME',0
     4E 41 4D 45 00
3666            11950 NRESP   EQU     $-1
3667 EB31       11960         DW      NPARM+1
3669 73         11970         DB      SW!STR!SGL!3,'MPW',0
     4D 50 57 00
366D            11980 MRESP   EQU     $-1
366E 3B32       11990         DW      MPARM+1
3670 44         12000         DB      SW!4,'SDEN',0
     53 44 45 4E 00
```

Format Init Code

```
3676 8331      12Ø1Ø          DW      SDPARM+1
3678 44        12Ø2Ø          DB      SW!4,'DDEN',Ø
     44 44 45 4E ØØ
367E 8831      12Ø3Ø          DW      DDPARM+1
368Ø 85        12Ø4Ø          DB      VAL!5,'SIDES',Ø
     53 49 44 45 53 ØØ
3687 8D31      12Ø5Ø          DW      SIDES+1
3689 93        12Ø6Ø          DB      VAL!SGL!3,'CYL',Ø
     43 59 4C ØØ
368E 9231      12Ø7Ø          DW      CPARM+1
369Ø 84        12Ø8Ø          DB      VAL!4,'STEP',Ø
     53 54 45 5Ø ØØ
3696 9731      12Ø9Ø          DW      STEPARM+1
3698 53        121ØØ          DB      SW!SGL!3,'ABS',Ø
     41 42 53 ØØ
369D 2635      1211Ø          DW      APARM+1
369F 55        1212Ø          DB      SW!SGL!5,'QUERY',Ø
     51 55 45 52 59 ØØ
36A6 B132      1213Ø          DW      QPARM+1
36A8 46        1214Ø          DB      SW!6,'SYSTEM',Ø
     53 59 53 54 45 4D ØØ
36BØ 1C26      1215Ø          DW      SYSPRM+1
36B2 94        1216Ø          DB      VAL!SGL!4,'WAIT',Ø
     57 41 49 54 ØØ
36B8 2C27      1217Ø          DW      WAITPRM+1
36BA 93        1218Ø          DB      VAL!SGL!3,'DIR',Ø
     44 49 52 ØØ
36BF 1228      1219Ø          DW      DIRPARM+1
36C1 ØØ        122ØØ          NOP
               1221Ø ;
36C2 46        1222Ø HELLO$   DB      'FORMAT'
     4F 52 4D 41 54
36C8           1223Ø *GET     CLIENT:3
               1224Ø ;CLIENTS/ASM - File to establish sign-on headers
               1225Ø ;
36C8 2Ø        1226Ø          DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
36F2 2Ø        1227Ø          DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
               1228Ø ;
37Ø7 41        1229Ø          DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
372F 2Ø        123ØØ          DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
3747 43        1231Ø HARD$    DB      'Cannot "SYSTEM" a floppy',CR
     61 6E 6E 6F 74 2Ø 22 53
```

Format Init Code

```
     59 53 54 45 4D 22 2Ø 61
     2Ø 66 6C 6F 7Ø 7Ø 79 ØD
376Ø 49         1232Ø NOMEM$  DB        'Insufficient memory for '
     6E 73 75 66 66 69 63 69
     65 6E 74 2Ø 6D 65 6D 6F
     72 79 2Ø 66 6F 72 2Ø
3778 73         1233Ø         DB        'specified format',CR
     7Ø 65 63 69 66 69 65 64
     2Ø 66 6F 72 6D 61 74 ØD
3789 57         1234Ø WHDRV$  DB        'Which drive is to be used ? ',3
     68 69 63 68 2Ø 64 72 69
     76 65 2Ø 69 73 2Ø 74 6F
     2Ø 62 65 2Ø 75 73 65 64
     2Ø 3F 2Ø Ø3
37A6 44         1235Ø DSKNAM$ DB        'Diskette name ? ',3
     69 73 6B 65 74 74 65 2Ø
     6E 61 6D 65 2Ø 3F 2Ø Ø3
37B7 4D         1236Ø MPW$    DB        'Master password ? ',3
     61 73 74 65 72 2Ø 7Ø 61
     73 73 77 6F 72 64 2Ø 3F
     2Ø Ø3
37CA 4E         1237Ø NUMCYL$ DB        'Number of cylinders ? ',3
     75 6D 62 65 72 2Ø 6F 66
     2Ø 63 79 6C 69 6E 64 65
     72 73 2Ø 3F 2Ø Ø3
37E1 42         1238Ø STEP5$  DB        'Boot strap stepping rate '
     6F 6F 74 2Ø 73 74 72 61
     7Ø 2Ø 73 74 65 7Ø 7Ø 69
     6E 67 2Ø 72 61 74 65 2Ø
37FA 3C         1239Ø         DB        '<6, 12, 2Ø, 3Ø msecs> ? ',3
     36 2C 2Ø 31 32 2C 2Ø 32
     3Ø 2C 2Ø 33 3Ø 2Ø 6D 73
     65 63 73 3E 2Ø 3F 2Ø Ø3
3813 42         124ØØ STEP8$  DB        'Bootstrap stepping rate '
     6F 6F 74 73 74 72 61 7Ø
     2Ø 73 74 65 7Ø 7Ø 69 6E
     67 2Ø 72 61 74 65 2Ø
382B 3C         1241Ø         DB        '<3, 6, 1Ø, 15/2Ø msecs> ? ',3
     33 2C 2Ø 36 2C 2Ø 31 3Ø
     2C 2Ø 31 35 2F 32 3Ø 2Ø
     6D 73 65 63 73 3E 2Ø 3F
     2Ø Ø3
3846 45         1242Ø SIDES$  DB        'Enter number of sides <1,2> ? ',3
     6E 74 65 72 2Ø 6E 75 6D
     62 65 72 2Ø 6F 66 2Ø 73
     69 64 65 73 2Ø 3C 31 2C
     32 3E 2Ø 3F 2Ø Ø3
3865 53         1243Ø DEN?$   DB        'Single or Double density <S,D> ? ',3
     69 6E 67 6C 65 2Ø 6F 72
     2Ø 44 6F 75 62 6C 65 2Ø
     64 65 6E 73 69 74 79 2Ø
     3C 53 2C 44 3E 2Ø 3F 2Ø
     Ø3
3887 44         1244Ø NOTRDY$ DB        'Drive not ready',CR
     72 69 76 65 2Ø 6E 6F 74
     2Ø 72 65 61 64 79 ØD
3897 57         1245Ø CANTWR$ DB        'Write protected disk',CR
     72 69 74 65 2Ø 7Ø 72 6F
     74 65 63 74 65 64 2Ø 64
```

Format Init Code

```
     69 73 6B ØD
38AC 44          1246Ø NODRV$  DB      'Drive not in system',CR
     72 69 76 65 2Ø 6E 6F 74
     2Ø 69 6E 2Ø 73 79 73 74
     65 6D ØD
38CØ 4C          1247Ø PMTDST$ DB      'Load destination diskette  <ENTER>',CR
     6F 61 64 2Ø 64 65 73 74
     69 6E 61 74 69 6F 6E 2Ø
     64 69 73 6B 65 74 74 65
     2Ø 2Ø 3C 45 4E 54 45 52
     3E ØD
38E3 44          1248Ø HASDAT$ DB      'Disk contains data -- ',3
     69 73 6B 2Ø 63 6F 6E 74
     61 69 6E 73 2Ø 64 61 74
     61 2Ø 2D 2D 2Ø Ø3
38FA 4E          1249Ø NOFMT$  DB      'Non-standard format',CR
     6F 6E 2D 73 74 61 6E 64
     61 72 64 2Ø 66 6F 72 6D
     61 74 ØD
39ØE 55          125ØØ CANTRD$ DB      'Unreadable directory',CR
     6E 72 65 61 64 61 62 6C
     65 2Ø 64 69 72 65 63 74
     6F 72 79 ØD
3923 4E          1251Ø NODIR$  DB      'Non-initialized directory',CR
     6F 6E 2D 69 6E 69 74 69
     61 6C 69 7A 65 64 2Ø 64
     69 72 65 63 74 6F 72 79
     ØD
393D 4E          1252Ø PACKID$ DB      'Name=XXXXXXXX  Date=MM/DD/YY',CR
     61 6D 65 3D 58 58 58 58
     58 58 58 58 2Ø 2Ø 44 61
     74 65 3D 4D 4D 2F 44 44
     2F 59 59 ØD
395A 2Ø          1253Ø OLDMPW$ DB      '  Enter its Master Password'
     2Ø 45 6E 74 65 72 2Ø 69
     74 73 2Ø 4D 61 73 74 65
     72 2Ø 5Ø 61 73 73 77 6F
     72 64
3975 2Ø          1254Ø         DB      ' or <BREAK> to abort: ',3
     6F 72 2Ø 3C 42 52 45 41
     4B 3E 2Ø 74 6F 2Ø 61 62
     6F 72 74 3A 2Ø Ø3
398C 2A          1255Ø LASTMSG DB      '*** The target drive is a hard disk ***',LF
     2A 2A 2Ø 54 68 65 2Ø 74
     61 72 67 65 74 2Ø 64 72
     69 76 65 2Ø 69 73 2Ø 61
     2Ø 68 61 72 64 2Ø 64 69
     73 6B 2Ø 2A 2A 2A ØA
39B4 41          1256Ø SURE?$  DB      'Are you sure you want to format it ? ',3
     72 65 2Ø 79 6F 75 2Ø 73
     75 72 65 2Ø 79 6F 75 2Ø
     77 61 6E 74 2Ø 74 6F 2Ø
     66 6F 72 6D 61 74 2Ø 69
     74 2Ø 3F 2Ø Ø3
39DA ØA          1257Ø INVMPW$ DB      LF,'Invalid Master Password',LF,CR
     49 6E 76 61 6C 69 64 2Ø
     4D 61 73 74 65 72 2Ø 5Ø
     61 73 73 77 6F 72 64 ØA
     ØD
```

Format Init Code

```
39F4 49          1258Ø BADNAM$ DB      'Invalid Disk Name',CR
     6E 76 61 6C 69 64 2Ø 44
     69 73 6B 2Ø 4E 61 6D 65
     ØD
3AØ6 44          1259Ø PAKNAM$ DB      'DATADISK'
     41 54 41 44 49 53 4B
3AØE 5Ø          126ØØ PAKMPW$ DB      'PASSWORD'
     41 53 53 57 4F 52 44
                 Ø852Ø ;
3A16             Ø853Ø         SUBTTL  <>
31ØØ             Ø854Ø         END     FORMAT
```

```
@@1             0000 @@2              0000 @@3              0000
@@4             0000 @MOD2            0000 @MOD4            FFFF
AFLOP           27D3 APARM            3525 BADMPW           3645
BADNAM          3641 BADNAM$          39F4 BFMT1            26A4
BFMT2           26A7 BFMT3            26CF BFMT4            26D5
BFMT5           26FE BGNFMT           2698 BGNVER           2735
BOOT            2F00 BOOTDIR          2A9D BOOTST$          2600
BREAK           29B9 BVER1            2747 BVER10           27B2
BVER3           2767 BVER4            2768 BVER5            2771
BVER8           278A BVER9            27A9 CALC1            2823
CALC2           2836 CALC3            283A CALCDIR          2806
CALCGPC         33E1 CANTRD$          390E CANTWR$          3897
CGPC1           33F4 CGPC2            33FF CGPC3            3407
CHGDEN          32F5 CKDR1            35F8 CKDR2            35FD
CKDR2A          3602 CKDR6            3608 CKDR7            3613
CKDRV           35EC CKMPW            35B5 CKMPW0           35BC
CKMPW1          35C3 CKMPW2           35D1 CKMPW3           35D9
CKMPW5          35DD CKMPW7           35E1 CKNAME           322B
CKWAIT          2724 CODE1            265F CODE1A           2661
CODF1           2669 CODF1A           266C CODF2            2670
CODF2A          2677 CODF3            267A CODF4            2682
CODF5           2657 CODRET           2665 CORE$            2F00
CPARM           3191 CR               000D CRT3             3C00
CRT4            F800 CVB1             3584 CVBIN            3582
CVD1            2A52 CVD2             2A59 CVDEC            2A50
CYLGRN          2951 D5TBL            2B2A D8TBL            2BC6
DATADSK$        4329 DCTCYL           3621 DCTLP1           313A
DDPARM          3187 DEN?$            3865 DFTMPW           3245
DFTNAM          31F5 DIRASC$          2CC6 DIRCYL$          2CA1
DIRDIR          2ABD DIRPARM          2811 DIRSET           2821
DRVNOP          29F6 DSKNAM           3218 DSKNAM$          37A6
DSPCYL          2A89 ERREXIT          29C5 EXIT             29C8
EXIT2           29DF EXIT3            29ED EXIT4            29F4
EXTERR          29BC FILLBLK          35AE FMT1             3144
FMT2            316D FMT2A            3175 FMT2B            31E4
FMTABT          29B9 FMTABT$          2D24 FMTCAO$          2D0E
FMTCYL$         2C39 FMTDAT           2639 FMTDRV           2A2A
FMTG$           2C6E FMTHD            2A13 FMTTBL           2608
FORMAT          3100 FORMATA          3109 GATBUF           2E00
GDDEN1          32D7 GENSYS           2840 GET3             2A5D
GET8            2A65 GET8A            2A6B GETDAT           3234
GETDEN          32B8 GETSYS2          3654 GETUC            2A77
GETUC1          2A82 GOFMT            2601 GOTSTEP          33DD
GSDEN1          32E7 GSYS1            28F2 GSYS2            2981
GSYS3           298B HARD$            3747 HASDAT$          38E3
HASHMPW         35B9 HELLO$           36C2 HITBUF           3000
HRDRV           27C1 HRDRV1           27E5 INITDEN          328C
INPMPW          3595 INVMPW           35E5 INVMPW$          39DA
IOERR           29A5 IPLSYS$          2CC9 LASTMSG          398C
LF              000A LSIID            28DB MOVFREE          2794
MOVMPW          324D MOVMPW1          325C MOVNAM           31FD
MOVNAM1         3211 MPARM            323A MPW              3263
MPW$            37B7 MPWBUF           3657 MRESP            366D
NOCYL$          2C7C NODIR            281E NODIR$           3923
NODRV$          38AC NOFMT$           38FA NOMEM$           3760
NOTFMT$         2D35 NOTHARD          3649 NOTRDY$          3887
NPARM           31EA NRESP            3666 NUMCYL$          37CA
OLDMPW          3508 OLDMPW$          395A PACKID$          393D
```

```
PAKMPW$       3AØE PAKNAM$      3AØ6 PASSWORD     42EØ
PCYL1         333Ø PCYL2        333C PCYL3        3341
PCYL4         334F PFMT1        3432 PFMT2        3479
PFMT3         348F PFMT4        34CD PFMT5        3522
PFMT6         3545 PMTCYL       3329 PMTDST       3412
PMTDST$       38CØ PMTS1        331E PMTSIDE      32F8
PMTSTEP       3363 PMTSYS$      2CEF PRMERR       364F
PRMMRG        32AB PRMTBL$      366Ø PRSMPW       326B
PSTEP1        33D6 QPARM        32BØ RDSEC        2A22
RESTOR        29FF RETCOD       29C9 RLS          ØØ62
RSELCT        2AØ9 S5TBL        2AEA S8TBL        2B81
SAFESP        44ØØ SDPARM       3182 SECCYL       2AE8
SECSKEW       26A1 SECTRK       2AE9 SELECT       29FA
SETDDEN       32DE SETSDEN      32EE SETSTD       32A4
SETUP         3559 SGL          ØØ1Ø SIDES        318C
SIDES$        3846 SPSAV        29D8 STAR$        2C69
STEP5         3382 STEP5$       37E1 STEP8        33AC
STEP8$        3813 STEPARM      3196 STEPDFT      2AE7
STEPIN        2AØ4 STOP         4326 STR          ØØ2Ø
STRLEN        ØØ17 SURE?$       39B4 SW           ØØ4Ø
SYSDCT        2ADD SYSPRM       261B TBLDATA      3629
TMPDCT        3617 TRKSKEW      27Ø2 TSTSID       3323
VAL           ØØ8Ø VERCYL$      2C51 VERS1        2A3C
VERSEC        2A27 VERSKEW      2764 VERSYS       2A31
WAITPRM       272B WHDRV        3151 WHDRV$       3789
WRCYL         2AØE WRDIR        2A41 WRDIR1       2A44
WRGAT1        28DF WRSEC        2A18 WRSYS        2A1D
@@ABORT       BE7D @@ADTSK      BF1Ø @@BANK       C428
@@BKSP        C1Ø8 @@BREAK      C43E @@CHNIO      BE68
@@CKBRKC      C48C @@CKDRV      BF64 @@CKEOF      C11D
@@CKTSK       BEFB @@CLOSE      CØF3 @@CLS        C476
@@CMNDI       BEA7 @@CMNDR      BEBC @@CTL        BCCC
@@DATE        BE3E @@DCSTAT     BFA3 @@DEBUG      BEE6
@@DECHEX      C3A8 @@DIRRD      C315 @@DIRWR      C32A
@@DIV16       C393 @@DIV8       C37E @@DODIR      BF79
@@DSP         BC9Ø @@DSPLY      BD3Ø @@ERROR      BED1
@@EXIT        BE92 @@FEXT       C282 @@FLAGS      C412
@@FNAME       C297 @@FSPEC      C26D @@GATRD      C3ØØ
@@GATWR       C33F @@GET        BCA4 @@GTDCB      C2C1
@@GTDCT       C2AC @@GTMOD      C2D6 @@HDFMT      CØ4B
@@HEX16       C3E7 @@HEX8       C3D2 @@HEXDEC     C3BD
@@HIGH$       C3FC @@INIT       CØC9 @@KBD        BDØ8
@@KEY         BC7C @@KEYIN      BD1C @@KLTSK      BF4F
@@LOAD        C243 @@LOC        C132 @@LOF        C147
@@LOGER       BD67 @@LOGOT      BD7C @@MSG        BDB3
@@MUL16       C369 @@MUL8       C354 @@OPEN       CØDE
@@PARAM       BE29 @@PAUSE      BE14 @@PEOF       C15C
@@POSN        C171 @@PRINT      BDC8 @@PRT        BCEØ
@@PUT         BCB8 @@RAMDIR     BF8E @@RDSEC      CØ21
@@RDSSC       C2EB @@READ       C186 @@REMOV      CØB4
@@RENAM       CØ9F @@REW        C19B @@RMTSK      BF25
@@RPTSK       BF3A @@RREAD      C1BØ @@RSLCT      CØØC
@@RSTOR       BFCD @@RUN        C258 @@RWRIT      C1C5
@@SEEK        BFF7 @@SEEKSC     C1DA @@SKIP       C1EF
@@SLCT        BFB8 @@STEPI      BFE2 @@TIME       BE53
@@VDCTL       BDFF @@VER        C2Ø4 @@VRSEC      CØ36
@@WEOF        C219 @@WHERE      BCF4 @@WRITE      C22E
@@WRSEC       CØ6Ø @@WRSSC      CØ75 @@WRTRK      CØ8A
```

```
ØØØØØ Total errors
```

NOTES:

NOTES:

FORMS/FLT - Printer output formmating filter

The Forms filter allows formatting of data sent to the *PR device. It is installed with the SET and FILTER Library commands. Its parameters are adjusted with the FORMS Library command.

```
               ØØ1ØØ ;FORMS/ASM - Printer Formatting Filter
ØØØØ           ØØ11Ø          TITLE   <FORMS/FLT - LS-DOS 6.2>
               ØØ12Ø ;
ØØØA           ØØ13Ø LF       EQU     1Ø
ØØØD           ØØ14Ø CR       EQU     13
               ØØ15Ø ;
ØØØØ           ØØ16Ø *GET     SVCMAC:3                ;SVC Macro equivalents
               ØØØ1Ø ;SVCMAC/ASM - LS-DOS Version VI
               ØØØ2Ø *LIST    OFF
               Ø39ØØ *LIST    ON
ØØØØ           ØØ17Ø *GET     COPYCOM:3               ;Copyright message
               Ø392Ø ; COPYCOM - File for Copyright COMment block
               Ø393Ø ;
ØØØØ           Ø394Ø          COM     '<*(C) 1982,83,84 by LSI*>'
               ØØ18Ø ;
24ØØ           ØØ19Ø          ORG     24ØØH
               ØØ2ØØ ;
               ØØ21Ø BEGIN
24ØØ           ØØ22Ø          @@CKBRKC                ;Check for break
24ØØ 3E6A      ØØØØ1          LD      A,1Ø6
24Ø2 EF        ØØØØ2          RST     4Ø
24Ø3 28Ø4      ØØ23Ø          JR      Z,BEGINA        ;Continue if no Break
24Ø5 21FFFF    ØØ24Ø          LD      HL,-1
24Ø8 C9        ØØ25Ø          RET                     ;  else abort
               ØØ26Ø ;
24Ø9 D5        ØØ27Ø BEGINA   PUSH    DE              ;Save DCB address
24ØA DDE1      ØØ28Ø          POP     IX              ;  in index reg
24ØC ED534B26  ØØ29Ø          LD      (PFDCB),DE      ;  and in filter header
241Ø           ØØ3ØØ          @@DSPLY HELLO$          ;Welcome the user
               ØØØØ3          IFEQ    Ø1H,1
241Ø 212525    ØØØØ4          LD      HL,HELLO$
               ØØØØ5          ENDIF
2413 3EØA      ØØØØ6          LD      A,1Ø
2415 EF        ØØØØ7          RST     4Ø
               ØØ31Ø ;
               ØØ32Ø ;        Check if entry from SET command
               ØØ33Ø ;
2416           ØØ34Ø          @@FLAGS                 ;IY => flag table base
2416 3E65      ØØØØ8          LD      A,1Ø1
2418 EF        ØØØØ9          RST     4Ø
2419 FDCBØ25E  ØØ35Ø          BIT     3,(IY+'C'-'A')  ;System request?
241D CAØ525    ØØ36Ø          JP      Z,VIASET        ;Quit if not
               ØØ37Ø ;
               ØØ38Ø ;        Check if filter is already resident
               ØØ39Ø ;
242Ø 112125    ØØ4ØØ          LD      DE,FF$          ;Check if filter is
2423           ØØ41Ø          @@GTMOD                 ;  already resident
2423 3E53      ØØØ1Ø          LD      A,83
2425 EF        ØØØ11          RST     4Ø
2426 EB        ØØ42Ø          EX      DE,HL           ;Put DCB ptr to HL
2427 2Ø23      ØØ43Ø          JR      NZ,NOTRES       ;Go if not
               ØØ44Ø ;
               ØØ45Ø ;        Make sure that the new DCB is same as the old
               ØØ46Ø ;
2429 ED4B4B26  ØØ47Ø          LD      BC,(PFDCB)      ;Replace DCB pointer
242D 79        ØØ48Ø          LD      A,C             ;  with new one
242E 4E        ØØ49Ø          LD      C,(HL)          ;P/u DCB pointer LSB
242F ØØ        ØØ5ØØ          NOP
243Ø 23        ØØ51Ø          INC     HL
2431 78        ØØ52Ø          LD      A,B
2432 46        ØØ53Ø          LD      B,(HL)          ;P/u DCB pointer MSB
```

```
2433 210600    00540          LD      HL,6            ;Get old DCB name &
2436 09        00550          ADD     HL,BC           ;  stuff into error
2437 7E        00560          LD      A,(HL)          ;  message in case
2438 2C        00570          INC     L               ;  a different DCB
2439 66        00580          LD      H,(HL)          ;  is referenced
243A 6F        00590          LD      L,A
243B 22FB25    00600          LD      (DCBNAM$),HL    ;Stuff message with spec
243E B4        00610          OR      H
243F 2876      00620          JR      Z,ISRES
2441 2A4B26    00630          LD      HL,(PFDCB)      ;P/u DCB existing DCB
2444 B7        00640          OR      A               ;  pointer
2445 ED42      00650          SBC     HL,BC           ;Same DCB pointer?
2447 C20925    00660          JP      NZ,DCBERR       ;Can't install if diff
244A 186B      00670          JR      ISRES
               00680 ;
               00690 ;        Module is not resident
               00700 ;
244C 114B49    00710 NOTRES   LD      DE,'IK'
244F           00720          @@GTDCB                 ;Locate low memory ptr
244F 3E52      00012          LD      A,82
2451 EF        00013          RST     40
2452 C21725    00730          JP      NZ,IOERR        ;Quit if not found
2455 2D        00740          DEC     L
2456 56        00750          LD      D,(HL)          ;P/u pointer to
2457 2D        00760          DEC     L               ;  start of free
2458 5E        00770          LD      E,(HL)          ;  low core
2459 ED53A724  00780          LD      (LCPTR+1),DE    ;Save loc for later
245D E5        00790          PUSH    HL              ;Save low core ptr
245E 210101    00800          LD      HL,PFEND-PFFLT
2461 19        00810          ADD     HL,DE           ;Start + driver length
2462 E5        00820          PUSH    HL
2463 2B        00830          DEC     HL              ;Point to last byte
2464 22DD24    00840          LD      (SVEND+1),HL
2467 010013    00850          LD      BC,1300H        ;Max addr + 1
246A AF        00860          XOR     A
246B ED42      00870          SBC     HL,BC
246D D1        00880          POP     DE              ;Rcvr new lc
246E E1        00890          POP     HL              ;Rcvr low core ptr
246F 382F      00900          JR      C,PUTLOW        ;If room, put low
               00910 ;
               00920 ;        Check if high memory available
               00930 ;
2471           00940          @@FLAGS
2471 3E65      00014          LD      A,101
2473 EF        00015          RST     40
2474 FDCB0246  00950          BIT     0,(IY+'C'-'A')  ;Memory frozen?
2478 C20D25    00960          JP      NZ,NOROOM       ;"No memory...
247B 210000    00970          LD      HL,0            ;Get HIGH$
247E 45        00980          LD      B,L
247F           00990          @@HIGH$
247F 3E64      00016          LD      A,100
2481 EF        00017          RST     40
2482 22DD24    01000          LD      (SVEND+1),HL    ;Save for relocator
2485 5D        01010          LD      E,L             ;Xfer new last
2486 54        01020          LD      D,H             ;  to reg DE
2487 AF        01030          XOR     A               ;Calc new start
2488 010101    01040          LD      BC,PFEND-PFFLT  ;BC = filter len
248B ED42      01050          SBC     HL,BC
248D 0600      01060          LD      B,0
248F           01070          @@HIGH$                 ;Set new HIGH$
248F 3E64      00018          LD      A,100
```

```
2491 EF        ØØØ19          RST      4Ø
2492 23        Ø1Ø8Ø          INC      HL              ;Point to new start
2493 EB        Ø1Ø9Ø          EX       DE,HL
2494 D5        Ø11ØØ          PUSH     DE
2495 CDD624    Ø111Ø          CALL     RELO            ;Relocate internal references
2498 D1        Ø112Ø          POP      DE
2499 3EFF      Ø113Ø          LD       A,ØFFH
249B 32C824    Ø114Ø          LD       (HGHFLG),A      ;Flag to notify user
249E 18Ø9      Ø115Ø          JR       MOVMOD          ;  himem used
               Ø116Ø ;
               Ø117Ø ;        Room in low core - move driver low
               Ø118Ø ;
24AØ 73        Ø119Ø PUTLOW   LD       (HL),E          ;Stuff low core ptr
24A1 2C        Ø12ØØ          INC      L               ;  with new low
24A2 72        Ø121Ø          LD       (HL),D
24A3 CDD624    Ø122Ø          CALL     RELO            ;Relocate vectors
24A6 11ØØØØ    Ø123Ø LCPTR    LD       DE,$-$          ;Low core pointer
               Ø124Ø ;
               Ø125Ø ;        Move module to memory
               Ø126Ø ;
24A9 D5        Ø127Ø MOVMOD   PUSH     DE              ;Save start
24AA 214326    Ø128Ø          LD       HL,PFFLT
24AD Ø1Ø1Ø1    Ø129Ø          LD       BC,PFEND-PFFLT  ;Calc driver length
24BØ EDBØ      Ø13ØØ          LDIR
24B2 D1        Ø131Ø          POP      DE              ;Pop filter start
24B3 FDCBØ3EE  Ø132Ø          SET      5,(IY+'D'-'A')  ;Set PF in DFLAG$
               Ø133Ø ;
24B7 21FE25    Ø134Ø ISRES    LD       HL,PFACT$       ;Init "FORMS installed
24BA DD36ØØ47  Ø135Ø          LD       (IX),4ØH!7      ;Init DCB type to "C/P/G"
24BE DD73Ø1    Ø136Ø          LD       (IX+1),E        ;  & filter & stuff the
24C1 DD72Ø2    Ø137Ø          LD       (IX+2),D        ;  filter address
24C4           Ø138Ø          @@LOGOT                  ;Display installation
               ØØØ2Ø          IFEQ     ØØH,1
               ØØØ21          LD       HL,
               ØØØ22          ENDIF
24C4 3EØC      ØØØ23          LD       A,12
24C6 EF        ØØØ24          RST      4Ø
24C7 3EØØ      Ø139Ø          LD       A,$-$
24C8           Ø14ØØ HGHFLG   EQU      $-1             ;Flag filter went high
24C9 B7        Ø141Ø          OR       A               ;Skip if not set
24CA 28Ø6      Ø142Ø          JR       Z,NTHGH
24CC 211B26    Ø143Ø          LD       HL,HMEM$        ;  else show "Went in himem
24CF           Ø144Ø          @@LOGOT
               ØØØ25          IFEQ     ØØH,1
               ØØØ26          LD       HL,
               ØØØ27          ENDIF
24CF 3EØC      ØØØ28          LD       A,12
24D1 EF        ØØØ29          RST      4Ø
24D2 21ØØØØ    Ø145Ø NTHGH    LD       HL,Ø            ;No error
24D5 C9        Ø146Ø          RET                      ;Done, back to user
               Ø147Ø ;
               Ø148Ø ;        Relocate internal references in driver
               Ø149Ø ;
24D6 DDE5      Ø15ØØ RELO     PUSH     IX
24D8 DD214427  Ø151Ø          LD       IX,RELTAB       ;Point to relocation tbl
24DC 21ØØØØ    Ø152Ø SVEND    LD       HL,$-$          ;Find distance to move
24DF 224526    Ø153Ø          LD       (PFFLT+2),HL    ;Set last byte used
24E2 114327    Ø154Ø          LD       DE,PFEND-1
24E5 B7        Ø155Ø          OR       A               ;Clear carry flag
24E6 ED52      Ø156Ø          SBC      HL,DE
24E8 44        Ø157Ø          LD       B,H             ;Move to BC
```

```
24E9 4D        Ø158Ø        LD      C,L
24EA 3EØE      Ø159Ø        LD      A,TABLEN        ;Get table length
24EC DD6EØØ    Ø16ØØ RLOOP  LD      L,(IX)          ;Get address to change
24EF DD66Ø1    Ø161Ø        LD      H,(IX+1)
24F2 5E        Ø162Ø        LD      E,(HL)          ;P/U address
24F3 23        Ø163Ø        INC     HL
24F4 56        Ø164Ø        LD      D,(HL)
24F5 EB        Ø165Ø        EX      DE,HL           ;Offset it
24F6 Ø9        Ø166Ø        ADD     HL,BC
24F7 EB        Ø167Ø        EX      DE,HL
24F8 72        Ø168Ø        LD      (HL),D          ;Put it back
24F9 2B        Ø169Ø        DEC     HL
24FA 73        Ø17ØØ        LD      (HL),E
24FB DD23      Ø171Ø        INC     IX
24FD DD23      Ø172Ø        INC     IX
24FF 3D        Ø173Ø        DEC     A
25ØØ 2ØEA      Ø174Ø        JR      NZ,RLOOP        ;Loop till done
25Ø2 DDE1      Ø175Ø        POP     IX
25Ø4 C9        Ø176Ø        RET
               Ø177Ø ;
               Ø178Ø ;      Error exits
               Ø179Ø ;
25Ø5 21BØ25    Ø18ØØ VIASET LD      HL,VIASET$      ;"Install with Set
25Ø8 DD        Ø181Ø        DB      ØDDH
25Ø9 21DF25    Ø182Ø DCBERR LD      HL,DCBERR$      ;"Filter in use
25ØC DD        Ø183Ø        DB      ØDDH
25ØD 21C525    Ø184Ø NOROOM LD      HL,NOROOM$      ;"Memory frozen
251Ø           Ø185Ø        @@LOGOT                 ;Show the error
               ØØØ3Ø        IFEQ    ØØH,1
               ØØØ31        LD      HL,
               ØØØ32        ENDIF
251Ø 3EØC      ØØØ33        LD      A,12
2512 EF        ØØØ34        RST     4Ø
2513 21FFFF    Ø186Ø        LD      HL,-1           ;Set abort code
2516 C9        Ø187Ø        RET
               Ø188Ø ;
2517 6F        Ø189Ø IOERR  LD      L,A             ;Error # to HL
2518 26ØØ      Ø19ØØ        LD      H,Ø
251A F6CØ      Ø191Ø        OR      ØCØH            ;Abbrev, return
251C 4F        Ø192Ø        LD      C,A             ;Error code to C
251D           Ø193Ø        @@ERROR                 ;  for error display
251D 3E1A      ØØØ35        LD      A,26
251F EF        ØØØ36        RST     4Ø
252Ø C9        Ø194Ø        RET
               Ø195Ø ;
               Ø196Ø ;      Messages & Data tables
               Ø197Ø ;
2521 24        Ø198Ø FF$    DB      '$FF',3
     46 46 Ø3
2525 46        Ø199Ø HELLO$ DB      'FORMS Filter'
     4F 52 4D 53 2Ø 46 69 6C
     74 65 72
2531           Ø2ØØØ *GET   CLIENT:3
               Ø395Ø ;CLIENTS/ASM - File to establish sign-on headers
               Ø396Ø ;
2531 2Ø        Ø397Ø        DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
255B 2Ø          Ø398Ø          DB       ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                 Ø399Ø ;
257Ø 41          Ø4ØØØ          DB       'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
2598 2Ø          Ø4Ø1Ø          DB       ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
                 Ø2Ø1Ø ;
25BØ 4D          Ø2Ø2Ø VIASET$ DB       'Must install via SET',CR
     75 73 74 2Ø 69 6E 73 74
     61 6C 6C 2Ø 76 69 61 2Ø
     53 45 54 ØD
25C5 4E          Ø2Ø3Ø NOROOM$ DB       'No memory space available',CR
     6F 2Ø 6D 65 6D 6F 72 79
     2Ø 73 7Ø 61 63 65 2Ø 61
     76 61 69 6C 61 62 6C 65
     ØD
25DF 46          Ø2Ø4Ø DCBERR$ DB       'Filter already attached to *xx',CR
     69 6C 74 65 72 2Ø 61 6C
     72 65 61 64 79 2Ø 61 74
     74 61 63 68 65 64 2Ø 74
     6F 2Ø 2A 78 78 ØD
25FB             Ø2Ø5Ø DCBNAM$ EQU      $-3
25FE 46          Ø2Ø6Ø PFACT$  DB       'Forms filter is now resident',CR
     6F 72 6D 73 2Ø 66 69 6C
     74 65 72 2Ø 69 73 2Ø 6E
     6F 77 2Ø 72 65 73 69 64
     65 6E 74 ØD
261B ØA          Ø2Ø7Ø HMEM$   DB       LF,'Note: filter installed in high memory.',CR
     4E 6F 74 65 3A 2Ø 66 69
     6C 74 65 72 2Ø 69 6E 73
     74 61 6C 6C 65 64 2Ø 69
     6E 2Ø 68 69 67 68 2Ø 6D
     65 6D 6F 72 79 2E ØD
                 Ø2Ø8Ø ;
                 Ø2Ø9Ø ;
                 Ø21ØØ ;        Printer Filter - PF
                 Ø211Ø ;        Provides hard or soft form feed, line wraparound,
                 Ø212Ø ;        automatic form feeds between pages, tabs, blank
                 Ø213Ø ;        lines, 1 byte translation table, left margins,
                 Ø214Ø ;        and set-top-of-form character.
                 Ø215Ø ;
                 Ø216Ø *MOD
ØØØ3             Ø217Ø PFBIT   EQU      3                  ;Position in DFLAG
ØØØØ             Ø218Ø SPLBIT  EQU      Ø                  ;Position in DFLAG
ØØØA             Ø219Ø LF      EQU      1Ø
ØØØD             Ø22ØØ CR      EQU      13
                 Ø221Ø ;
2643 1814        Ø222Ø PFFLT   JR       PFBGN              ;Branch around header
2645 4327        Ø223Ø         DW       PFEND-1            ;Last byte used
2647 Ø3          Ø224Ø         DB       3,'$FF'            ;Name length/name
     24 46 46
```

```
264B ØØØØ       Ø225Ø PFDCB   DW      $-$             ;Link to DCB
264D ØØØØ       Ø226Ø         DW      Ø
                Ø227Ø ;
                Ø228Ø ;       Filter data area
                Ø229Ø ;
264F            Ø23ØØ PFDATA$ EQU     $
ØØØØ            Ø231Ø PMAX    EQU     $-PFDATA$
264F 42         Ø232Ø         DB      66              ;Page size (max lines per page)
ØØØ1            Ø233Ø LCOUNT  EQU     $-PFDATA$
265Ø ØØ         Ø234Ø         DB      Ø               ;Line counter
ØØØ2            Ø235Ø LMAX    EQU     $-PFDATA$
2651 42         Ø236Ø         DB      66              ;Max lines to print
ØØØ3            Ø237Ø CCOUNT  EQU     $-PFDATA$
2652 ØØ         Ø238Ø         DB      Ø               ;Chars per line printed
ØØØ4            Ø239Ø XL1     EQU     $-PFDATA$
2653 ØØ         Ø24ØØ         DB      Ø               ;Translate from
ØØØ5            Ø241Ø XL2     EQU     $-PFDATA$
2654 ØØ         Ø242Ø         DB      Ø               ;Translate to
ØØØ6            Ø243Ø INDENT  EQU     $-PFDATA$
2655 ØØ         Ø244Ø         DB      Ø               ;Indent after line wraparound
ØØØ7            Ø245Ø ADDLF   EQU     $-PFDATA$
2656 Ø4         Ø246Ø         DB      4               ;Bit-Ø, LF after CR; bit-1=FF
                Ø247Ø                                 ;Bit-2, TAB expand (1)
ØØØ8            Ø248Ø CMAX    EQU     $-PFDATA$
2657 ØØ         Ø249Ø         DB      Ø               ;Max CPL before wraparound
ØØØ9            Ø25ØØ MARGIN  EQU     $-PFDATA$
2658 ØØ         Ø251Ø         DB      Ø               ;Left hand margin
                Ø252Ø ;
                Ø253Ø ;       Start of filter
                Ø254Ø ;
2659 281A       Ø255Ø PFBGN   JR      Z,FFENTRY       ;Go if @PUT
265B 11         Ø256Ø         DB      Ø11H            ;Ignore next inst if not
265C Ø6Ø2       Ø257Ø PFPUT   LD      B,2             ;Init for @PUT
265E DDE5       Ø258Ø         PUSH    IX
266Ø DD2A4B26   Ø259Ø         LD      IX,(PFDCB)      ;Grab the DCB vector
2662            Ø26ØØ RXØ1    EQU     $-2
2664            Ø261Ø         @@CHNIO                 ;  & chain to it
2664 3E14       ØØØ37         LD      A,2Ø
2666 EF         ØØØ38         RST     4Ø
2667 DDE1       Ø262Ø         POP     IX
2669 C9         Ø263Ø         RET
                Ø264Ø ;
                Ø265Ø ;       Peform the tab function
                Ø266Ø ;
266A DD7EØ3     Ø267Ø DOTAB   LD      A,(IX+CCOUNT)   ;How many spaces to
266D E6Ø7       Ø268Ø         AND     7               ;  next tab stop?
266F D6Ø8       Ø269Ø         SUB     8
2671 ED44       Ø27ØØ         NEG
2673 1867       Ø271Ø         JR      @INDENT         ;Space over to it
                Ø272Ø ;
                Ø273Ø ;       Filter code
                Ø274Ø ;
2675 DD214F26   Ø275Ø FFENTRY LD      IX,PFDATA$      ;Base register
2677            Ø276Ø RXØ2    EQU     $-2
                Ø277Ø ;
2679 DD7EØ4     Ø278Ø CKXLAT  LD      A,(IX+XL1)      ;Get xlate in
267C B9         Ø279Ø         CP      C               ;Translate this char?
267D 2ØØ4       Ø28ØØ         JR      NZ,CONT         ;Go if not xlated char
267F DD7EØ5     Ø281Ø         LD      A,(IX+XL2)      ;Xlated to this
2682 4F         Ø282Ø         LD      C,A
2683 79         Ø283Ø CONT    LD      A,C             ;P/u char to test
```

```
2684 FEØC       Ø284Ø         CP      ØCH               ;Form feed?
2686 CA1B27     Ø285Ø         JP      Z,DOTOF
2687            Ø286Ø RX14    EQU     $-2
2689 FEØ6       Ø287Ø         CP      6                 ;SET TOF?
268B CA3E27     Ø288Ø         JP      Z,SETTOF
268C            Ø289Ø RXØ3    EQU     $-2
268E FEØD       Ø29ØØ         CP      CR                ;CR?
269Ø 287A       Ø291Ø         JR      Z,DOCRLF
2692 FEØA       Ø292Ø         CP      LF                ;LF?
2694 2876       Ø293Ø         JR      Z,DOCRLF
2696 DD7EØ9     Ø294Ø         LD      A,(IX+MARGIN)     ;Left margin to do?
2699 B7         Ø295Ø         OR      A
269A 28ØB       Ø296Ø         JR      Z,NOMARG          ;Go if not
269C DD34Ø3     Ø297Ø         INC     (IX+CCOUNT)       ;Check current char count
269F DD35Ø3     Ø298Ø         DEC     (IX+CCOUNT)       ;If at newline,
26A2 C5         Ø299Ø         PUSH    BC
26A3 CCDC26     Ø3ØØØ         CALL    Z,@INDENT         ;  need a margin now
26A4            Ø3Ø1Ø RX13    EQU     $-2
26A6 C1         Ø3Ø2Ø         POP     BC
26A7 79         Ø3Ø3Ø NOMARG  LD      A,C               ;P/u character again
26A8 DDCBØ756   Ø3Ø4Ø         BIT     2,(IX+ADDLF)      ;Expand tabs?
26AC 28Ø4       Ø3Ø5Ø         JR      Z,CONTA
26AE FEØ9       Ø3Ø6Ø         CP      9                 ;Tab?
26BØ 28B8       Ø3Ø7Ø         JR      Z,DOTAB
26B2 FE2Ø       Ø3Ø8Ø CONTA   CP      2ØH               ;Other control code?
26B4 38A6       Ø3Ø9Ø         JR      C,PFPUT           ;Pass on unchanged if so
                Ø31ØØ ;
                Ø311Ø ;       Got a character to output
                Ø312Ø ;
26B6 C5         Ø313Ø PUTCHAR PUSH    BC                ;Save character
26B7 CDCØ26     Ø314Ø         CALL    SETUP             ;Setup for next char
26B8            Ø315Ø RX12    EQU     $-2
26BA C1         Ø316Ø         POP     BC
26BB CØ         Ø317Ø         RET     NZ                ;Quit on error
26BC CC5C26     Ø318Ø         CALL    Z,PFPUT           ;Now put the char
26BD            Ø319Ø RXØ4    EQU     $-2
26BF C9         Ø32ØØ         RET
                Ø321Ø ;
                Ø322Ø ;       Do the end of line check
                Ø323Ø ;
26CØ DD34Ø3     Ø324Ø SETUP   INC     (IX+CCOUNT)       ;Inc char counter
26C3 DD7EØ8     Ø325Ø         LD      A,(IX+CMAX)       ;Wraparound needed?
26C6 A7         Ø326Ø         AND     A
26C7 C8         Ø327Ø         RET     Z                 ;Quit if feature is off
26C8 DDBEØ3     Ø328Ø         CP      (IX+CCOUNT)
26CB 3Ø75       Ø329Ø         JR      NC,EXITZ          ;Done if not needed
26CD CDØC27     Ø33ØØ         CALL    DOCRLF            ;Do carriage return
26CE            Ø331Ø RXØ5    EQU     $-2
26DØ CØ         Ø332Ø         RET     NZ
26D1 DD34Ø3     Ø333Ø         INC     (IX+CCOUNT)       ;Adjust char counter
                Ø334Ø ;
                Ø335Ø ;       Check on indent needed
                Ø336Ø ;
26D4 DD7EØ6     Ø337Ø         LD      A,(IX+INDENT)     ;P/u indent
26D7 DD86Ø9     Ø338Ø         ADD     A,(IX+MARGIN)     ;Add in the MARGIN
26DA B7         Ø339Ø         OR      A
26DB C8         Ø34ØØ         RET     Z                 ;Done if none
26DC C5         Ø341Ø @INDENT PUSH    BC                ;In case of recursive
26DD 47         Ø342Ø         LD      B,A               ;  calls
26DE ØE2Ø       Ø343Ø         LD      C,' '             ;Print spaces
26EØ C5         Ø344Ø SPACES  PUSH    BC                ;Save counter
```

```
26E1 AF         Ø345Ø          XOR     A
26E2 CDB626     Ø346Ø          CALL    PUTCHAR         ;Put the character
26E3            Ø347Ø RXØ6     EQU     $-2
26E5 C1         Ø348Ø          POP     BC              ;Recover counter
26E6 2ØØ2       Ø349Ø          JR      NZ,$+4          ;Exit on PUT error
26E8 1ØF6       Ø35ØØ          DJNZ    SPACES
26EA C1         Ø351Ø          POP     BC
26EB C9         Ø352Ø          RET
26EC DDCBØ746   Ø353Ø LINFEED  BIT     Ø,(IX+ADDLF)
26FØ 28Ø8       Ø354Ø          JR      Z,DOWN1         ;Go if hardware auto-LF
26F2 ØEØD       Ø355Ø          LD      C,CR            ;Else do CR and LF
26F4 CD5C26     Ø356Ø          CALL    PFPUT
26F5            Ø357Ø RX11     EQU     $-2
26F7 CØ         Ø358Ø          RET     NZ
26F8 18Ø8       Ø359Ø          JR      DOWNLF
26FA DD7EØ3     Ø36ØØ DOWN1    LD      A,(IX+CCOUNT)
26FD A7         Ø361Ø          AND     A               ;Line empty?
26FE ØEØD       Ø362Ø          LD      C,CR            ;Do CR if not
27ØØ 2ØØ2       Ø363Ø          JR      NZ,DOWNCR
27Ø2 ØEØA       Ø364Ø DOWNLF   LD      C,LF            ;Do LF if so
27Ø4 CD5C26     Ø365Ø DOWNCR   CALL    PFPUT
27Ø5            Ø366Ø RXØ7     EQU     $-2
27Ø7 DD36Ø3ØØ   Ø367Ø          LD      (IX+CCOUNT),Ø   ;Starting new line
27ØB C9         Ø368Ø          RET
                Ø369Ø ;
27ØC CDEC26     Ø37ØØ DOCRLF   CALL    LINFEED         ;CRLF & check if page end
27ØD            Ø371Ø RXØ8     EQU     $-2
27ØF CØ         Ø372Ø          RET     NZ
                Ø373Ø ;
271Ø DD34Ø1     Ø374Ø          INC     (IX+LCOUNT)
2713 DD7EØ1     Ø375Ø          LD      A,(IX+LCOUNT)   ;Time to do form feed?
2716 DDBEØ2     Ø376Ø          CP      (IX+LMAX)
2719 3827       Ø377Ø          JR      C,EXITZ         ;Return if not
                Ø378Ø ;
271B DD7EØØ     Ø379Ø DOTOF    LD      A,(IX+PMAX)     ;How many lines to feed?
271E DD96Ø1     Ø38ØØ          SUB     (IX+LCOUNT)
2721 281B       Ø381Ø          JR      Z,SETTOF        ;Skip if zero
2723 C5         Ø382Ø          PUSH    BC              ;In case called by DOTAB
2724 47         Ø383Ø          LD      B,A
2725 DDCBØ74E   Ø384Ø          BIT     1,(IX+ADDLF)    ;Hardware form feed?
2729 28Ø7       Ø385Ø          JR      Z,SOFTFF        ;Go if not
272B ØEØC       Ø386Ø          LD      C,ØCH           ;  else load up TOF char
272D CD5C26     Ø387Ø          CALL    PFPUT           ;  and send it
272E            Ø388Ø RXØ9     EQU     $-2
273Ø 18ØB       Ø389Ø          JR      FFEXIT
2732 C5         Ø39ØØ SOFTFF   PUSH    BC
2733 CDEC26     Ø391Ø          CALL    LINFEED         ;Do LF's
2734            Ø392Ø RX1Ø     EQU     $-2
2736 C1         Ø393Ø          POP     BC
2737 28Ø2       Ø394Ø          JR      Z,CHRGONE       ;This linefeed sent OK
2739 C1         Ø395Ø          POP     BC              ;  else clean stack
273A C9         Ø396Ø          RET                     ;  and return error
273B 1ØF5       Ø397Ø CHRGONE  DJNZ    SOFTFF
273D C1         Ø398Ø FFEXIT   POP     BC
                Ø399Ø ;
                Ø4ØØØ ;        Set the top-of-form
                Ø4Ø1Ø ;
273E DD36Ø1ØØ   Ø4Ø2Ø SETTOF   LD      (IX+LCOUNT),Ø   ;Reset line counter
2742 BF         Ø4Ø3Ø EXITZ    CP      A
2743 C9         Ø4Ø4Ø          RET
                Ø4Ø5Ø ;
```

```
2744           Ø4Ø6Ø PFEND   EQU     $
               Ø4Ø7Ø ;
2744 6226      Ø4Ø8Ø RELTAB  DW      RXØ1,RXØ2,RXØ3,RXØ4,RXØ5,RXØ6,RXØ7,RXØ8
     7726 8C26 BD26 CE26 E326 Ø527 ØD27
2754 2E27      Ø4Ø9Ø         DW      RXØ9,RX1Ø,RX11,RX12,RX13,RX14
     3427 F526 B826 A426 8726
ØØØE           Ø41ØØ TABLEN  EQU     $-RELTAB/2
               Ø411Ø ;
24ØØ           Ø412Ø         END     BEGIN
```

```
@@1             ØØØØ @@2             ØØØØ @@3             ØØØØ
@@4             ØØØØ @INDENT         26DC @MOD2           ØØØØ
@MOD4           FFFF ADDLF           ØØØ7 BEGIN           24ØØ
BEGINA          24Ø9 CCOUNT          ØØØ3 CHRGONE         273B
CKXLAT          2679 CMAX            ØØØ8 CONT            2683
CONTA           26B2 CR              ØØØD DCBERR          25Ø9
DCBERR$         25DF DCBNAM$         25FB DOCRLF          27ØC
DOTAB           266A DOTOF           271B DOWN1           26FA
DOWNCR          27Ø4 DOWNLF          27Ø2 EXITZ           2742
FF$             2521 FFENTRY         2675 FFEXIT          273D
HELLO$          2525 HGHFLG          24C8 HMEM$           261B
INDENT          ØØØ6 IOERR           2517 ISRES           24B7
LCOUNT          ØØØ1 LCPTR           24A6 LF              ØØØA
LINFEED         26EC LMAX            ØØØ2 MARGIN          ØØØ9
MOVMOD          24A9 NOMARG          26A7 NOROOM          25ØD
NOROOM$         25C5 NOTRES          244C NTHGH           24D2
PFACT$          25FE PFBGN           2659 PFBIT           ØØØ3
PFDATA$         264F PFDCB           264B PFEND           2744
PFFLT           2643 PFPUT           265C PMAX            ØØØØ
PUTCHAR         26B6 PUTLOW          24AØ RELO            24D6
RELTAB          2744 RLOOP           24EC RXØ1            2662
RXØ2            2677 RXØ3            268C RXØ4            26BD
RXØ5            26CE RXØ6            26E3 RXØ7            27Ø5
RXØ8            27ØD RXØ9            272E RX1Ø            2734
RX11            26F5 RX12            26B8 RX13            26A4
RX14            2687 SETTOF          273E SETUP           26CØ
SOFTFF          2732 SPACES          26EØ SPLBIT          ØØØØ
SVEND           24DC TABLEN          ØØØE VIASET          25Ø5
VIASET$         25BØ XL1             ØØØ4 XL2             ØØØ5
@@ABORT         8FAE @@ADTSK         9Ø41 @@BANK          9559
@@BKSP          9239 @@BREAK         956F @@CHNIO         8F99
@@CKBRKC        95BD @@CKDRV         9Ø95 @@CKEOF         924E
@@CKTSK         9Ø2C @@CLOSE         9224 @@CLS           95A7
@@CMNDI         8FD8 @@CMNDR         8FED @@CTL           8DFD
@@DATE          8F6F @@DCSTAT        9ØD4 @@DEBUG         9Ø17
@@DECHEX        94D9 @@DIRRD         9446 @@DIRWR         945B
@@DIV16         94C4 @@DIV8          94AF @@DODIR         9ØAA
@@DSP           8DC1 @@DSPLY         8E61 @@ERROR         9ØØ2
@@EXIT          8FC3 @@FEXT          93B3 @@FLAGS         9543
@@FNAME         93C8 @@FSPEC         939E @@GATRD         9431
@@GATWR         947Ø @@GET           8DD5 @@GTDCB         93F2
@@GTDCT         93DD @@GTMOD         94Ø7 @@HDFMT         917C
@@HEX16         9518 @@HEX8          95Ø3 @@HEXDEC        94EE
@@HIGH$         952D @@INIT          91FA @@KBD           8E39
@@KEY           8DAD @@KEYIN         8E4D @@KLTSK         9Ø8Ø
@@LOAD          9374 @@LOC           9263 @@LOF           9278
@@LOGER         8E98 @@LOGOT         8EAD @@MSG           8EE4
@@MUL16         949A @@MUL8          9485 @@OPEN          92ØF
@@PARAM         8F5A @@PAUSE         8F45 @@PEOF          928D
@@POSN          92A2 @@PRINT         8EF9 @@PRT           8E11
@@PUT           8DE9 @@RAMDIR        9ØBF @@RDSEC         9152
@@RDSSC         941C @@READ          92B7 @@REMOV         91E5
@@RENAM         91DØ @@REW           92CC @@RMTSK         9Ø56
@@RPTSK         9Ø6B @@RREAD         92E1 @@RSLCT         913D
@@RSTOR         9ØFE @@RUN           9389 @@RWRIT         92F6
@@SEEK          9128 @@SEEKSC        93ØB @@SKIP          932Ø
@@SLCT          9ØE9 @@STEPI         9113 @@TIME          8F84
@@VDCTL         8F3Ø @@VER           9335 @@VRSEC         9167
@@WEOF          934A @@WHERE         8E25 @@WRITE         935F
@@WRSEC         9191 @@WRSSC         91A6 @@WRTRK         91BB
```

```
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

KSM/FLT - Keystroke multiply filter

The KSM filter allows multiple characters or lines to be assigned to an alphabetic key. It must be installed with the SET and FILTER Library commands. It will install in high memory, and will not attempt to use the low driver zone.

```
                ØØ1ØØ ;KSM/ASM - Keystroke Multiply Filter
ØØØØ            ØØ11Ø         TITLE   <KSM/FLT - LS-DOS 6.2>
                ØØ12Ø ;
ØØØA            ØØ13Ø LF      EQU     1Ø
ØØØD            ØØ14Ø CR      EQU     13
                ØØ15Ø ;
ØØØØ            ØØ16Ø *GET    SVCMAC:3                ;SVC Macro equivalents
                ØØØ1Ø ;SVCMAC/ASM - LS-DOS Version VI
                ØØØ2Ø *LIST   OFF
                Ø39ØØ *LIST   ON
ØØØØ            ØØ17Ø *GET    COPYCOM:3               ;Copyright message
                Ø392Ø ; COPYCOM - File for Copyright COMment block
                Ø393Ø ;
ØØØØ            Ø394Ø         COM     '<*(C) 1982,83,84 by LSI*>'
                ØØ18Ø ;
24ØØ            ØØ19Ø         ORG     24ØØH
                ØØ2ØØ ;
                ØØ21Ø KSM
24ØØ            ØØ22Ø         @@CKBRKC                ;Check for break
24ØØ 3E6A       ØØØØ1         LD      A,1Ø6
24Ø2 EF         ØØØØ2         RST     4Ø
24Ø3 28Ø4       ØØ23Ø         JR      Z,KSMA          ;Contine if not
24Ø5 21FFFF     ØØ24Ø         LD      HL,-1           ;  else abort
24Ø8 C9         ØØ25Ø         RET
                ØØ26Ø ;
24Ø9 ED53CA27   ØØ27Ø KSMA    LD      (KSMDCB),DE     ;Save ptr to DCB
24ØD E5         ØØ28Ø         PUSH    HL              ;Save ptr to cmdline buf
24ØE            ØØ29Ø         @@DSPLY HELLO$          ;Display copyright msg
                ØØØØ3         IFEQ    Ø1H,1
24ØE 217225     ØØØØ4         LD      HL,HELLO$
                ØØØØ5         ENDIF
2411 3EØA       ØØØØ6         LD      A,1Ø
2413 EF         ØØØØ7         RST     4Ø
2414            ØØ3ØØ         @@FLAGS                 ;Get flags
2414 3E65       ØØØØ8         LD      A,1Ø1
2416 EF         ØØØØ9         RST     4Ø
2417 E1         ØØ31Ø         POP     HL              ;Rcvr cmndline pointer
                ØØ32Ø ;
                ØØ33Ø ;       Check if entry from SET command
                ØØ34Ø ;
2418 FDCBØ25E   ØØ35Ø         BIT     3,(IY+'C'-'A')  ;System request?
241C CA4B25     ØØ36Ø         JP      Z,VIASET        ;Quit if not
241F 11A126     ØØ37Ø         LD      DE,KSMFCB       ;Point to FCB
2422            ØØ38Ø         @@FSPEC                 ;Fetch the KSM filespec
2422 3E4E       ØØØ1Ø         LD      A,78
2424 EF         ØØØ11         RST     4Ø
2425 C25725     ØØ39Ø         JP      NZ,SPCREQ       ;Jump on bad spec
2428 D5         ØØ4ØØ         PUSH    DE              ;Save FCB pointer
2429 119626     ØØ41Ø         LD      DE,PRMTBL$      ;Load param table pointer
242C            ØØ42Ø         @@PARAM                 ;Parse parms
242C 3E11       ØØØ12         LD      A,17
242E EF         ØØØ13         RST     4Ø
242F D1         ØØ43Ø         POP     DE              ;Recover FCB pointer
243Ø C26325     ØØ44Ø         JP      NZ,IOERR        ;Go on parm error
2433 217225     ØØ45Ø         LD      HL,DFTKSM       ;Init to default ext
2436            ØØ46Ø         @@FEXT                  ;Fetch if not entered
2436 3E4F       ØØØ14         LD      A,79
2438 EF         ØØØ15         RST     4Ø
                ØØ47Ø ;
                ØØ48Ø ;       Transfer requested ENTER char to test loc
                ØØ49Ø ;
```

```
2439 213B00     00500 EPARM     LD      HL,';'          ; set default ";"
243C 3A9D26     00510           LD      A,(ERSP)        ;Test parm response
243F CB77       00520           BIT     6,A             ;Flag is no good!
2441 C26125     00530           JP      NZ,PRMERR
2444 CB6F       00540           BIT     5,A             ;Test string or value
2446 7E         00550           LD      A,(HL)          ;P/u assumed string
2447 2001       00560           JR      NZ,$+3          ;Go if string entry
2449 7D         00570           LD      A,L             ;P/u hex or dec entry
244A 321328     00580           LD      (ECHAR+1),A     ;Stuff it in there
244D D5         00590           PUSH    DE
244E 116D25     00600           LD      DE,KSM$         ;Check if filter is
2451            00610           @@GTMOD                 ;  already resident
2451 3E53       00016           LD      A,83
2453 EF         00017           RST     40
2454 227D24     00620           LD      (KSMMEM+1),HL   ;Stuff start
2457 EB         00630           EX      DE,HL           ;Put DCB ptr to HL
2458 D1         00640           POP     DE
2459 202C       00650           JR      NZ,OPENKSM      ;Go if not
                00660 ;
                00670 ;         Make sure that the new DCB is same as the old
                00680 ;
245B E5         00690           PUSH    HL              ;Save where to stuff
245C 4E         00700           LD      C,(HL)          ;P/u DCB pointer LSB
245D 23         00710           INC     HL
245E 46         00720           LD      B,(HL)          ;P/u DCB pointer MSB
245F 210600     00730           LD      HL,6            ;Get old DCB name &
2462 09         00740           ADD     HL,BC           ;  stuff into error
2463 7E         00750           LD      A,(HL)          ;  message in case
2464 2C         00760           INC     L               ;  a different DCB
2465 66         00770           LD      H,(HL)          ;  is referenced
2466 6F         00780           LD      L,A
2467 227226     00790           LD      (DCBNAM$),HL
246A B4         00800           OR      H               ;If DCB name is null,
246B 2ACA27     00810           LD      HL,(KSMDCB)
246E E5         00820           PUSH    HL              ;Save pointer to stuff
246F 2803       00830           JR      Z,UPDPTR        ;  then OK to use
2471 B7         00840           OR      A
2472 ED42       00850           SBC     HL,BC           ;Same DCB pointer?
2474 C1         00860 UPDPTR    POP     BC              ;Rcvr pointer to stuff
2475 E1         00870           POP     HL              ;Rcvr address to put pointer
2476 C24F25     00880           JP      NZ,DCBERR       ;Quit if filter in use
                00890 ;
                00900 ;         Same DCB - Okay to stuff
                00910 ;
2479 71         00920           LD      (HL),C          ;Store the DCB pointer
247A 23         00930           INC     HL
247B 70         00940           LD      (HL),B
247C 210000     00950 KSMMEM    LD      HL,$-$          ;If res, ptr to start
247F 015200     00960           LD      BC,ECHAR-DVRBGN+1
2482 09         00970           ADD     HL,BC           ;Resident, stuff ECHAR
2483 3A1328     00980           LD      A,(ECHAR+1)     ;  where it is in memory
2486 77         00990           LD      (HL),A          ;Stuff in upper mem
2487 21C126     01000 OPENKSM   LD      HL,KSMBUF       ;Pt to buffer area
248A 0600       01010           LD      B,0             ;Init LRL=256
248C            01020           @@OPEN                  ;Open the file
248C 3E3B       00018           LD      A,59
248E EF         00019           RST     40
248F C26325     01030           JP      NZ,IOERR        ;Jump on open error
2492 212128     01040           LD      HL,DVREND       ;Place file in memory 1st
2495 061A       01050           LD      B,26            ;Init for 26 lines
2497            01060 KSM1      @@GET                   ;Get a char from file
```

```
2497 3EØ3      ØØØ2Ø          LD      A,3
2499 EF        ØØØ21          RST     4Ø
249A 2ØØC      Ø1Ø7Ø          JR      NZ,KSM2         ;Jump on error
249C 77        Ø1Ø8Ø          LD      (HL),A          ;Stuff into memory
249D 23        Ø1Ø9Ø          INC     HL              ;Inc memory pointer
249E FEØD      Ø11ØØ          CP      CR              ;Found end-of-line?
24AØ 2ØF5      Ø111Ø          JR      NZ,KSM1         ;Loop if not
24A2 1ØF3      Ø112Ø          DJNZ    KSM1            ;Decrement the A-Z loop
24A4 2B        Ø113Ø          DEC     HL              ;Backup over last CR &
24A5 Ø4        Ø114Ø          INC     B               ;  adjust for one more
24A6 3E1C      Ø115Ø          LD      A,1CH           ;No error here, just EOF
24A8 F5        Ø116Ø KSM2     PUSH    AF              ;Save error code
24A9           Ø117Ø          @@CLOSE                 ;Close the file
24A9 3E3C      ØØØ22          LD      A,6Ø
24AB EF        ØØØ23          RST     4Ø
24AC F1        Ø118Ø          POP     AF
24AD FE1C      Ø119Ø          CP      1CH             ;Ck for eof
24AF C26325    Ø12ØØ          JP      NZ,IOERR        ;Jump on not eof error
24B2 36ØD      Ø121Ø KSM3     LD      (HL),CR         ;End with a <ENTER>
24B4 23        Ø122Ø          INC     HL              ;For all remaining
24B5 1ØFB      Ø123Ø          DJNZ    KSM3            ;"letters" not entered
24B7 DD2ACA27  Ø124Ø          LD      IX,(KSMDCB)     ;Rcvr user DCB entry
24BB 112128    Ø125Ø          LD      DE,DVREND       ;Calculate the length
24BE AF        Ø126Ø          XOR     A               ; of the KSM file just
24BF ED52      Ø127Ø          SBC     HL,DE           ; loaded
24C1 44        Ø128Ø          LD      B,H             ;Xfer length
24C2 4D        Ø129Ø          LD      C,L
24C3 2A7D24    Ø13ØØ          LD      HL,(KSMMEM+1)   ;If not previously res,
24C6 7D        Ø131Ø          LD      A,L             ;  move to HIGH$
24C7 B4        Ø132Ø          OR      H
24C8 281E      Ø133Ø          JR      Z,MOVTOHI
24CA C5        Ø134Ø          PUSH    BC              ;Save length
24CB E5        Ø135Ø          PUSH    HL              ;Save old start
24CC Ø9        Ø136Ø          ADD     HL,BC           ;Start + data
24CD 3812      Ø137Ø          JR      C,KSM3A         ;Bad if wrap past Ø
24CF Ø161ØØ    Ø138Ø          LD      BC,DVREND-DVRBGN+1
24D2 Ø9        Ø139Ø          ADD     HL,BC           ;Start + data + filter
24D3 38ØC      Ø14ØØ          JR      C,KSM3A         ;Bad if wrap past Ø
24D5 EB        Ø141Ø          EX      DE,HL           ;Save in reg DE
24D6 E1        Ø142Ø          POP     HL              ;Rcvr old start
24D7 23        Ø143Ø          INC     HL              ;Pt to last byte used
24D8 23        Ø144Ø          INC     HL
24D9 7E        Ø145Ø          LD      A,(HL)          ;P/u last byte used
24DA 23        Ø146Ø          INC     HL
24DB 66        Ø147Ø          LD      H,(HL)          ;  into HL
24DC 6F        Ø148Ø          LD      L,A
24DD E5        Ø149Ø          PUSH    HL
24DE AF        Ø15ØØ          XOR     A               ;Clear carry flag
24DF ED52      Ø151Ø          SBC     HL,DE           ;Is req > available?
24E1 E1        Ø152Ø KSM3A    POP     HL              ;Rcvr old start to reuse
24E2 C1        Ø153Ø          POP     BC              ;Rcvr length of req
24E3 DA5325    Ø154Ø          JP      C,NOROOM        ;Quit if file too big
24E6 18Ø9      Ø155Ø          JR      KSMØA
24E8 C5        Ø156Ø MOVTOHI  PUSH    BC              ;Save data length
24E9 21ØØØØ    Ø157Ø          LD      HL,Ø            ;P/u current high memory
24EC 45        Ø158Ø          LD      B,L
24ED           Ø159Ø          @@HIGH$
24ED 3E64      ØØØ24          LD      A,1ØØ
24EF EF        ØØØ25          RST     4Ø
24FØ C1        Ø16ØØ          POP     BC              ;Recover data length
24F1 22C327    Ø161Ø KSMØA    LD      (DVRBGN+2),HL   ;Stuff last byte used
```

```
24F4 22CF27    Ø162Ø          LD      (RX1),HL        ;Stuff ptr to flag byte
24F7 36ØØ      Ø163Ø          LD      (HL),Ø          ;Init the KSM char ptr
24F9 2B        Ø164Ø          DEC     HL              ;  to zero to show no
24FA 36ØØ      Ø165Ø          LD      (HL),Ø          ;  char avail at startup
24FC 2B        Ø166Ø          DEC     HL
24FD 112128    Ø167Ø          LD      DE,DVREND       ;Move data to high
25ØØ 1A        Ø168Ø MOVLP    LD      A,(DE)          ;Data is in reverse order
25Ø1 77        Ø169Ø          LD      (HL),A
25Ø2 2B        Ø17ØØ          DEC     HL              ;Dec himem ptr
25Ø3 13        Ø171Ø          INC     DE              ;  and inc the char ptr
25Ø4 ØB        Ø172Ø          DEC     BC              ;Reduce char count
25Ø5 78        Ø173Ø          LD      A,B             ;  and check if done
25Ø6 B1        Ø174Ø          OR      C
25Ø7 2ØF7      Ø175Ø          JR      NZ,MOVLP        ;Loop back if not
25Ø9 Ø16ØØØ    Ø176Ø          LD      BC,DVREND-DVRBGN        ;Get driver len
25ØC AF        Ø177Ø          XOR     A               ;Reduce potential HIGH$
25ØD ED42      Ø178Ø          SBC     HL,BC           ;  by driver length
25ØF 3A7D24    Ø179Ø          LD      A,(KSMMEM+1)    ;Don't update HIGH$
2512 B7        Ø18ØØ          OR      A               ;  if previously res
2513 28Ø9      Ø181Ø          JR      Z,DOHIGH        ;Go if not resident
               Ø182Ø ;
               Ø183Ø ;        Module already resident
               Ø184Ø ;
2515 ED5B7D24  Ø185Ø          LD      DE,(KSMMEM+1)   ;P/u module entry point
2519 213926    Ø186Ø          LD      HL,KSMRPL$      ;  & reuse the filter
251C 1818      Ø187Ø          JR      KSM8
               Ø188Ø ;
               Ø189Ø ;        Stuff new HIGH$ value (Note: B=Ø for driver
               Ø19ØØ ;        length so there is no damage on the @@HIGH$ SVC
               Ø191Ø ;
251E Ø6ØØ      Ø192Ø DOHIGH   LD      B,Ø
252Ø           Ø193Ø          @@HIGH$
252Ø 3E64      ØØØ26          LD      A,1ØØ
2522 EF        ØØØ27          RST     4Ø
2523 23        Ø194Ø          INC     HL              ;Pt to driver start
2524 EB        Ø195Ø          EX      DE,HL
2525 D5        Ø196Ø          PUSH    DE              ;Save start of driver
2526 21Ø9ØØ    Ø197Ø          LD      HL,KSMDCB-DVRBGN
2529 19        Ø198Ø          ADD     HL,DE           ;Point to filter DCB ptr
252A 22DF27    Ø199Ø          LD      (RX2),HL
252D 21C127    Ø2ØØØ          LD      HL,DVRBGN       ;Move parms also
253Ø EDBØ      Ø2Ø1Ø          LDIR
2532 D1        Ø2Ø2Ø          POP     DE              ;Rcvr driver ept
2533 212226    Ø2Ø3Ø          LD      HL,KSMACT$      ;Init "KSM installed
2536 DD36ØØ45  Ø2Ø4Ø KSM8     LD      (IX),4ØH!5      ;Init DCB type to "input"
253A DD73Ø1    Ø2Ø5Ø          LD      (IX+1),E        ;  & filter & stuff the
253D DD72Ø2    Ø2Ø6Ø          LD      (IX+2),D        ;  filter address
254Ø FDCBØ3F6  Ø2Ø7Ø          SET     6,(IY+'D'-'A')  ;Turn on device flag bit
2544           Ø2Ø8Ø          @@LOGOT                 ;Display installation msg
               ØØØ28          IFEQ    ØØH,1
               ØØØ29          LD      HL,
               ØØØ3Ø          ENDIF
2544 3EØC      ØØØ31          LD      A,12
2546 EF        ØØØ32          RST     4Ø
2547 21ØØØØ    Ø2Ø9Ø          LD      HL,Ø            ;Set no error
254A C9        Ø21ØØ          RET                     ;Back to the user
               Ø211Ø ;
               Ø212Ø ;        Error processing
               Ø213Ø ;
254B 21FB25    Ø214Ø VIASET   LD      HL,VIASET$      ;"Install with Set
254E DD        Ø215Ø          DB      ØDDH
```

```
254F 215226     Ø216Ø DCBERR  LD      HL,DCBERR$      ;"Filter in use already
2552 DD         Ø217Ø         DB      ØDDH
2553 217526     Ø218Ø NOROOM  LD      HL,NOROOM$      ;"Memory frozen
2556 DD         Ø219Ø         DB      ØDDH
2557 211Ø26     Ø22ØØ SPCREQ  LD      HL,SPCREQ$      ;"Missing filespec
255A            Ø221Ø         @@LOGOT                 ;Display an error
                ØØØ33         IFEQ    ØØH,1
                ØØØ34         LD      HL,
                ØØØ35         ENDIF
255A 3EØC       ØØØ36         LD      A,12
255C EF         ØØØ37         RST     4Ø
255D 21FFFF     Ø222Ø         LD      HL,-1           ;Set abort code
256Ø C9         Ø223Ø         RET
2561 3E2C       Ø224Ø PRMERR  LD      A,44            ;Init PARM ERROR
2563 6F         Ø225Ø IOERR   LD      L,A             ;Error code to HL
2564 26ØØ       Ø226Ø         LD      H,Ø
2566 F6CØ       Ø227Ø         OR      ØCØH            ;Set short, return
2568 4F         Ø228Ø         LD      C,A             ;Error to C
2569            Ø229Ø         @@ERROR                 ;  for error display
2569 3E1A       ØØØ38         LD      A,26
256B EF         ØØØ39         RST     4Ø
256C C9         Ø23ØØ         RET
                Ø231Ø ;
                Ø232Ø ;       Data and message area
                Ø233Ø ;
256D 24         Ø234Ø KSM$    DB      '$KSM',3
     4B 53 4D Ø3
2572            Ø235Ø DFTKSM  EQU     $               ;Note: HELLO$ must follow
2572 4B         Ø236Ø HELLO$  DB      'KSM Filter'
     53 4D 2Ø 46 69 6C 74 65
     72
257C            Ø237Ø *GET    CLIENT:3
                Ø395Ø ;CLIENTS/ASM - File to establish sign-on headers
                Ø396Ø ;
257C 2Ø         Ø397Ø         DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
25A6 2Ø         Ø398Ø         DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                Ø399Ø ;
25BB 41         Ø4ØØØ         DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
25E3 2Ø         Ø4Ø1Ø         DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
                Ø238Ø ;
25FB 4D         Ø239Ø VIASET$ DB      'Must install via SET',CR
     75 73 74 2Ø 69 6E 73 74
     61 6C 6C 2Ø 76 69 61 2Ø
     53 45 54 ØD
```

```
261Ø 46         Ø24ØØ SPCREQ$ DB      'Filespec required',CR
     69 6C 65 73 7Ø 65 63 2Ø
     72 65 71 75 69 72 65 64
     ØD
2622 4B         Ø241Ø KSMACT$ DB      'KSM is now operational',CR
     53 4D 2Ø 69 73 2Ø 6E 6F
     77 2Ø 6F 7Ø 65 72 61 74
     69 6F 6E 61 6C ØD
2639 4B         Ø242Ø KSMRPL$ DB      'KSM filter data replaced',CR
     53 4D 2Ø 66 69 6C 74 65
     72 2Ø 64 61 74 61 2Ø 72
     65 7Ø 6C 61 63 65 64 ØD
2652 4B         Ø243Ø DCBERR$ DB      'KSM filter already attached to *xx',CR
     53 4D 2Ø 66 69 6C 74 65
     72 2Ø 61 6C 72 65 61 64
     79 2Ø 61 74 74 61 63 68
     65 64 2Ø 74 6F 2Ø 2A 78
     78 ØD
2672            Ø244Ø DCBNAM$ EQU     $-3
2675 52         Ø245Ø NOROOM$ DB      'Request exceeds available memory',CR
     65 71 75 65 73 74 2Ø 65
     78 63 65 65 64 73 2Ø 61
     76 61 69 6C 61 62 6C 65
     2Ø 6D 65 6D 6F 72 79 ØD
2696 D2         Ø246Ø PRMTBL$ DB      'R'!8ØH,ØF5H,'ENTER',Ø
     F5 45 4E 54 45 52 ØØ
269D            Ø247Ø ERSP    EQU     $-1
                Ø248Ø ;
269E 3A24       Ø249Ø         DW      EPARM+1
26AØ ØØ         Ø25ØØ         DB      Ø
                Ø251Ø ;
ØØ2Ø            Ø252Ø KSMFCB  DEFS    32
Ø1ØØ            Ø253Ø KSMBUF  DEFS    256
                Ø254Ø ;
                Ø255Ø ;       Key-Stroke Multiplication driver
                Ø256Ø ;
27C1 18ØB       Ø257Ø DVRBGN  JR      START           ;Branch around header
27C3 ØØØØ       Ø258Ø         DW      $-$             ;Last byte used
27C5 Ø4         Ø259Ø         DB      4,'$KSM'
     24 4B 53 4D
27CA ØØØØ       Ø26ØØ KSMDCB  DW      $-$             ;Pointer to KSM's DCB
27CC ØØØØ       Ø261Ø         DW      Ø
                Ø262Ø ;
27CE 21ØØØØ     Ø263Ø START   LD      HL,Ø            ;P/u possible address to
27CF            Ø264Ø RX1     EQU     $-2
27D1 56         Ø265Ø         LD      D,(HL)          ;  a KSM that was parsed
27D2 2B         Ø266Ø         DEC     HL              ;  to a ';' logical ENTER
27D3 5E         Ø267Ø         LD      E,(HL)          ;If this vector is zero,
27D4 2B         Ø268Ø         DEC     HL              ;  no KSM continuation is
27D5 EB         Ø269Ø         EX      DE,HL           ;  pending - find a new
27D6 F5         Ø27ØØ         PUSH    AF              ;  entry. Save flags.
27D7 7C         Ø271Ø         LD      A,H             ;  If <> Ø, grab the KSM
27D8 B5         Ø272Ø         OR      L               ;  line continuation
27D9 2Ø2B       Ø273Ø         JR      NZ,DVR4A
27DB F1         Ø274Ø         POP     AF              ;Rcvr flags
27DC D5         Ø275Ø         PUSH    DE              ;Save ptr to 'A'-KSM
27DD DD2ACA27   Ø276Ø DVR1    LD      IX,(KSMDCB)     ;Chain to next DCB module
27DF            Ø277Ø RX2     EQU     $-2
27E1            Ø278Ø         @@CHNIO
27E1 3E14       ØØØ4Ø         LD      A,2Ø
27E3 EF         ØØØ41         RST     4Ø
```

```
27E4 D1                    POP     DE              ;Rcvr 'A'-KSM pointer
27E5 C0                    RET     NZ              ;Back if nothing or error
27E6 CB7F                  BIT     7,A             ;Is it a CLEAR function?
27E8 C8                    RET     Z               ;Ret if <CLEAR> not down
27E9 F5                    PUSH    AF              ;Save key entry
27EA FEC1                  CP      'A'+80H         ;Ck for range A-Z
27EC 3804                  JR      C,DVR2          ;Exit if < 'A'
27EE FEDB                  CP      'Z'+1+80H
27F0 3803                  JR      C,DVR3          ;Use it if A-Z
27F2 F1            DVR2    POP     AF              ;Rcvr orig flag
27F3 BF                    CP      A               ;Set Z-flag
27F4 C9                    RET
                   ;
                   ;       Key code entry includes <CLEAR> key
                   ;
27F5 F1            DVR3    POP     AF              ;Rcvr orig flag
27F6 62                    LD      H,D             ;Rcvr ptr to 'A'-KSM
27F7 6B                    LD      L,E             ;  & xfer to reg HL
27F8 D6C1                  SUB     80H+'A'         ;Adjust offset to index
27FA 280B                  JR      Z,DVR5          ;Bypass if was 'A'
27FC 47                    LD      B,A             ;Set loop counter
27FD 3E0D                  LD      A,CR            ;Read past the KSM lines
27FF BE            DVR4    CP      (HL)            ;  for letters preceding
2800 2B                    DEC     HL              ;  key entry to find the
2801 20FC                  JR      NZ,DVR4         ;  KSM line for entered
2803 10FA                  DJNZ    DVR4            ;  key code
2805 3E                    DB      3EH             ;Ignore next inst
                   ;
                   ;       Routine to pick up the next KSM character
                   ;       & return it to the system KI request
                   ;
2806 F1            DVR4A   POP     AF              ;Clean the stack
2807 7E            DVR5    LD      A,(HL)          ;P/u the next KSM char
2808 2B                    DEC     HL              ;Dec pointer to next one
2809 EB                    EX      DE,HL           ;Put either a pointer to
280A 23                    INC     HL              ;  the next KSM char or
280B FE0D                  CP      CR              ;  if got last, zero the
280D 280B                  JR      Z,DVR6          ;  data pointer
280F 73                    LD      (HL),E          ;Stuff pointer to next
2810 23                    INC     HL              ;  character to fetch
2811 72                    LD      (HL),D
2812 FE3B          ECHAR   CP      ';'             ;Ck on logical line end
2814 2002                  JR      NZ,DVR7         ;  & convert to <ENTER>
2816 3E0D                  LD      A,CR            ;  if it was semi-colon
2818 BF            DVR7    CP      A               ;Tell the system we have
2819 C9                    RET                     ;  retrieved a char
                   ;
                   ;       Got the terminating X'0D' - Clear the pointer
                   ;
281A AF            DVR6    XOR     A               ;Clear the KSM char ptr
281B 77                    LD      (HL),A          ;  as next request is new
281C 23                    INC     HL
281D 77                    LD      (HL),A
281E FEFF                  CP      0FFH            ;Set NZ & A = 0
2820 C9                    RET
2821               DVREND  EQU     $
                   ;
2400                       END     KSM
```

```
@@1                ØØØØ @@2            ØØØØ @@3            ØØØØ
@@4                ØØØØ @MOD2          ØØØØ @MOD4          FFFF
CR                 ØØØD DCBERR         254F DCBERR$        2652
DCBNAM$            2672 DFTKSM         2572 DOHIGH         251E
DVR1               27DD DVR2           27F2 DVR3           27F5
DVR4               27FF DVR4A          28Ø6 DVR5           28Ø7
DVR6               281A DVR7           2818 DVRBGN         27C1
DVREND             2821 ECHAR          2812 EPARM          2439
ERSP               269D HELLO$         2572 IOERR          2563
KSM                24ØØ KSM$           256D KSMØA          24F1
KSM1               2497 KSM2           24A8 KSM3           24B2
KSM3A              24E1 KSM8           2536 KSMA           24Ø9
KSMACT$            2622 KSMBUF         26C1 KSMDCB         27CA
KSMFCB             26A1 KSMMEM         247C KSMRPL$        2639
LF                 ØØØA MOVLP          25ØØ MOVTOHI        24E8
NOROOM             2553 NOROOM$        2675 OPENKSM        2487
PRMERR             2561 PRMTBL$        2696 RX1            27CF
RX2                27DF SPCREQ         2557 SPCREQ$        261Ø
START              27CE UPDPTR         2474 VIASET         254B
VIASET$            25FB @@ABORT        8E86 @@ADTSK        8F19
@@BANK             9431 @@BKSP         9111 @@BREAK        9447
@@CHNIO            8E71 @@CKBRKC       9495 @@CKDRV        8F6D
@@CKEOF            9126 @@CKTSK        8FØ4 @@CLOSE        9ØFC
@@CLS              947F @@CMNDI        8EBØ @@CMNDR        8EC5
@@CTL              8CD5 @@DATE         8E47 @@DCSTAT       8FAC
@@DEBUG            8EEF @@DECHEX       93B1 @@DIRRD        931E
@@DIRWR            9333 @@DIV16        939C @@DIV8         9387
@@DODIR            8F82 @@DSP          8C99 @@DSPLY        8D39
@@ERROR            8EDA @@EXIT         8E9B @@FEXT         928B
@@FLAGS            941B @@FNAME        92AØ @@FSPEC        9276
@@GATRD            93Ø9 @@GATWR        9348 @@GET          8CAD
@@GTDCB            92CA @@GTDCT        92B5 @@GTMOD        92DF
@@HDFMT            9Ø54 @@HEX16        93FØ @@HEX8         93DB
@@HEXDEC           93C6 @@HIGH$        94Ø5 @@INIT         9ØD2
@@KBD              8D11 @@KEY          8C85 @@KEYIN        8D25
@@KLTSK            8F58 @@LOAD         924C @@LOC          913B
@@LOF              915Ø @@LOGER        8D7Ø @@LOGOT        8D85
@@MSG              8DBC @@MUL16        9372 @@MUL8         935D
@@OPEN             9ØE7 @@PARAM        8E32 @@PAUSE        8E1D
@@PEOF             9165 @@POSN         917A @@PRINT        8DD1
@@PRT              8CE9 @@PUT          8CC1 @@RAMDIR       8F97
@@RDSEC            9Ø2A @@RDSSC        92F4 @@READ         918F
@@REMOV            9ØBD @@RENAM        9ØA8 @@REW          91A4
@@RMTSK            8F2E @@RPTSK        8F43 @@RREAD        91B9
@@RSLCT            9Ø15 @@RSTOR        8FD6 @@RUN          9261
@@RWRIT            91CE @@SEEK         9ØØØ @@SEEKSC       91E3
@@SKIP             91F8 @@SLCT         8FC1 @@STEPI        8FEB
@@TIME             8E5C @@VDCTL        8EØ8 @@VER          92ØD
@@VRSEC            9Ø3F @@WEOF         9222 @@WHERE        8CFD
@@WRITE            9237 @@WRSEC        9Ø69 @@WRSSC        9Ø7E
@@WRTRK            9Ø93
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

LOG/CMD - Log in a disk

The main use of the Log utility is to allow swapping a double sided disk into drive Ø after booting on a single sided disk.

```
                 00100 ;LOG/ASM - Optional Disk Log Program
0000             00110          TITLE   <LOG - LS-DOS 6.2>
                 00120 ;
000D             00130 CR       EQU     13
000A             00140 LF       EQU     10
000E             00150 CRSON    EQU     14
                 00160 ;
0000             00170 *GET     SVCMAC:3                ;SVC Macro equivalents
                 00010 ;SVCMAC/ASM - LS-DOS Version VI
                 00020 *LIST    OFF
                 03900 *LIST    ON
0000             00180 *GET     COPYCOM:3               ;Copyright message
                 03920 ; COPYCOM - File for Copyright COMment block
                 03930 ;
0000             03940          COM     '<*(C) 1982,83,84 by LSI*>'
                 00190 ;
2600             00200          ORG     2600H
                 00210 ;
                 00220 LOG
2600             00230          @@CKBRKC                ;Check for break
2600 3E6A        00001          LD      A,106
2602 EF          00002          RST     40
2603 2804        00240          JR      Z,LOGA          ;Go if not
2605 21FFFF      00250          LD      HL,-1           ;  else abort
2608 C9          00260          RET
                 00270 ;
                 00280 LOGA
2609 ED738326    00290          LD      (STACK),SP      ;Save entry SP
260D E5          00300          PUSH    HL              ;Save cmdline ptr
260E             00310          @@DSPLY HELLO$          ;Display the signon msg
                 00003          IFEQ    01H,1
260E 219626      00004          LD      HL,HELLO$
                 00005          ENDIF
2611 3E0A        00006          LD      A,10
2613 EF          00007          RST     40
2614 E1          00320          POP     HL              ;Recover cmdline ptr
                 00330 ;
                 00340 ; Start of main module code
                 00350 ;
2615 0E00        00360 START    LD      C,0             ;Default drive 0
2617 7E          00370 SKIPSP   LD      A,(HL)          ;Scan command line
2618 23          00380          INC     HL
2619 FE20        00390          CP      ' '             ;Skip spaces
261B 28FA        00400          JR      Z,SKIPSP
261D FE3A        00410          CP      ':'             ;Look for colon
261F 200C        00420          JR      NZ,DEFALT       ;End of line if not found
2621 7E          00430          LD      A,(HL)          ;Get drive #
2622 D630        00440          SUB     30H             ;Make a number
2624 DA8926      00450          JP      C,ILLDRV        ;# too low
2627 FE08        00460          CP      7+1
2629 D28926      00470          JP      NC,ILLDRV       ;# too hi
262C 4F          00480          LD      C,A             ;Save in C
262D 79          00490 DEFALT   LD      A,C             ;Drive 0?
262E A7          00500          AND     A
262F 2018        00510          JR      NZ,NOWAIT       ;Go if not
2631             00520          @@DSPLY WAIT$           ;Display "Switch disks
                 00008          IFEQ    01H,1
2631 211E27      00009          LD      HL,WAIT$
                 00010          ENDIF
2634 3E0A        00011          LD      A,10
2636 EF          00012          RST     40
```

```
2637 2Ø52     ØØ53Ø          JR      NZ,IOERR
2639          ØØ54Ø          @@KEY                    ;Wait for a key
2639 3EØ1     ØØØ13          LD      A,1
263B EF       ØØØ14          RST     4Ø
263C 2Ø4D     ØØ55Ø          JR      NZ,IOERR
263E C5       ØØ56Ø          PUSH    BC               ;Save the drive #
263F ØEØD     ØØ57Ø          LD      C,CR             ;Output a new line
2641          ØØ58Ø          @@DSP
2641 3EØ2     ØØØ15          LD      A,2
2643 EF       ØØØ16          RST     4Ø
2644 C1       ØØ59Ø          POP     BC               ;Recover drive #
2645 2Ø44     ØØ6ØØ          JR      NZ,IOERR
2647 18Ø7     ØØ61Ø          JR      NOCHK            ;Can't call CKDRV if :Ø
              ØØ62Ø ;
2649          ØØ63Ø NOWAIT   @@CKDRV                  ;Drive ready?
2649 3E21     ØØØ17          LD      A,33
264B EF       ØØØ18          RST     4Ø
264C 3E2Ø     ØØ64Ø          LD      A,32             ;"Illegal drive number"
264E 2Ø3B     ØØ65Ø          JR      NZ,IOERR         ;Go if not ready
265Ø 21ØØ28   ØØ66Ø NOCHK    LD      HL,BUFFER        ;Sector buffer
2653 11ØØØØ   ØØ67Ø          LD      DE,Ø             ;Read boot sector
2656          ØØ68Ø          @@RDSEC
2656 3E31     ØØØ19          LD      A,49
2658 EF       ØØØ2Ø          RST     4Ø
2659 2Ø3Ø     ØØ69Ø          JR      NZ,IOERR         ;Go if error
265B          ØØ7ØØ          @@GTDCT                  ;Point IY to DCT
265B 3E51     ØØØ21          LD      A,81
265D EF       ØØØ22          RST     4Ø
265E 23       ØØ71Ø          INC     HL               ;Point HL to byte 2
265F 23       ØØ72Ø          INC     HL
266Ø 7E       ØØ73Ø          LD      A,(HL)           ;Get dir cyl #
2661 FD77Ø9   ØØ74Ø          LD      (IY+9),A         ;  and put in DCT
              ØØ75Ø ;
2664 57       ØØ76Ø          LD      D,A              ;Now read GAT
2665 21ØØ28   ØØ77Ø          LD      HL,BUFFER        ;Disk sector buffer
2668 5D       ØØ78Ø          LD      E,L              ;Set to Ø
2669          ØØ79Ø          @@RDSEC
2669 3E31     ØØØ23          LD      A,49
266B EF       ØØØ24          RST     4Ø
266C FEØ6     ØØ8ØØ          CP      6                ;Must be sys sector
266E 2Ø1B     ØØ81Ø          JR      NZ,IOERR         ;Go if error
              ØØ82Ø ;
267Ø 2ECD     ØØ83Ø          LD      L,ØCDH           ;Offset to disk type
2672 7E       ØØ84Ø          LD      A,(HL)           ;P/U disk type
2673 E62Ø     ØØ85Ø          AND     2ØH              ;Check # of sides bit
2675 47       ØØ86Ø          LD      B,A              ;Save in B
2676 FD7EØ4   ØØ87Ø          LD      A,(IY+4)         ;P/U byte in DCT
2679 E6DF     ØØ88Ø          AND     ØDFH             ;Mask out old value
267B BØ       ØØ89Ø          OR      B                ;Put in new value
267C FD77Ø4   ØØ9ØØ          LD      (IY+4),A         ;Put back in DCT
              ØØ91Ø ;
267F 21ØØØØ   ØØ92Ø          LD      HL,Ø             ;Set no error
2682 31ØØØØ   ØØ93Ø $QUIT    LD      SP,$-$           ;P/u original stack
2683          ØØ94Ø STACK    EQU     $-2
2685          ØØ95Ø          @@CKBRKC                 ;Clear any break
2685 3E6A     ØØØ25          LD      A,1Ø6
2687 EF       ØØØ26          RST     4Ø
2688 C9       ØØ96Ø          RET                      ;Back to the user
              ØØ97Ø ;
2689 3E2Ø     ØØ98Ø ILLDRV   LD      A,32             ;Init "illegal drv"
268B 6F       ØØ99Ø IOERR    LD      L,A              ;Put error # into HL
```

```
268C 26ØØ       Ø1ØØØ          LD      H,Ø
268E F6CØ       Ø1Ø1Ø          OR      ØCØH            ;Abbrev, return
269Ø 4F         Ø1Ø2Ø          LD      C,A             ;Error code to C
2691            Ø1Ø3Ø          @@ERROR                 ;  for error display
2691 3E1A       ØØØ27          LD      A,26
2693 EF         ØØØ28          RST     4Ø
2694 18EC       Ø1Ø4Ø          JR      $QUIT
                Ø1Ø5Ø ;
2696 4C         Ø1Ø6Ø HELLO$   DB      'LOG Drive'
     4F 47 2Ø 44 72 69 76 65
269F            Ø1Ø7Ø *GET    CLIENT:3
                Ø395Ø ;CLIENTS/ASM - File to establish sign-on headers
                Ø396Ø ;
269F 2Ø         Ø397Ø          DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
26C9 2Ø         Ø398Ø          DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                Ø399Ø ;
26DE 41         Ø4ØØØ          DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
27Ø6 2Ø         Ø4Ø1Ø          DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
271E 45         Ø1Ø8Ø WAIT$    DB      'Exchange disks and depress <ENTER> ',3
     78 63 68 61 6E 67 65 2Ø
     64 69 73 6B 73 2Ø 61 6E
     64 2Ø 64 65 7Ø 72 65 73
     73 2Ø 3C 45 4E 54 45 52
     3E 2Ø Ø3
28ØØ            Ø1Ø9Ø          ORG     $<-8+1<8
28ØØ            Ø11ØØ BUFFER   EQU     $
                Ø111Ø ;
26ØØ            Ø112Ø          END     LOG
```

```
$QUIT          2682 @@1           ØØØØ @@2            ØØØØ
@@3            ØØØØ @@4           ØØØØ @MOD2          ØØØØ
@MOD4          FFFF BUFFER        28ØØ CR             ØØØD
CRSON          ØØØE DEFALT        262D HELLO$         2696
ILLDRV         2689 IOERR         268B LF             ØØØA
LOG            26ØØ LOGA          26Ø9 NOCHK          265Ø
NOWAIT         2649 SKIPSP        2617 STACK          2683
START          2615 WAIT$         271E @@ABORT        75ØF
@@ADTSK        75A2 @@BANK        7ABA @@BKSP         779A
@@BREAK        7ADØ @@CHNIO       74FA @@CKBRKC       7B1E
@@CKDRV        75F6 @@CKEOF       77AF @@CKTSK        758D
@@CLOSE        7785 @@CLS         7BØ8 @@CMNDI        7539
@@CMNDR        754E @@CTL         735E @@DATE         74DØ
@@DCSTAT       7635 @@DEBUG       7578 @@DECHEX       7A3A
@@DIRRD        79A7 @@DIRWR       79BC @@DIV16        7A25
@@DIV8         7A1Ø @@DODIR       76ØB @@DSP          7322
@@DSPLY        73C2 @@ERROR       7563 @@EXIT         7524
@@FEXT         7914 @@FLAGS       7AA4 @@FNAME        7929
@@FSPEC        78FF @@GATRD       7992 @@GATWR        79D1
@@GET          7336 @@GTDCB       7953 @@GTDCT        793E
@@GTMOD        7968 @@HDFMT       76DD @@HEX16        7A79
@@HEX8         7A64 @@HEXDEC      7A4F @@HIGH$        7A8E
@@INIT         775B @@KBD         739A @@KEY          73ØE
@@KEYIN        73AE @@KLTSK       75E1 @@LOAD         78D5
@@LOC          77C4 @@LOF         77D9 @@LOGER        73F9
@@LOGOT        74ØE @@MSG         7445 @@MUL16        79FB
@@MUL8         79E6 @@OPEN        777Ø @@PARAM        74BB
@@PAUSE        74A6 @@PEOF        77EE @@POSN         78Ø3
@@PRINT        745A @@PRT         7372 @@PUT          734A
@@RAMDIR       762Ø @@RDSEC       76B3 @@RDSSC        797D
@@READ         7818 @@REMOV       7746 @@RENAM        7731
@@REW          782D @@RMTSK       75B7 @@RPTSK        75CC
@@RREAD        7842 @@RSLCT       769E @@RSTOR        765F
@@RUN          78EA @@RWRIT       7857 @@SEEK         7689
@@SEEKSC       786C @@SKIP        7881 @@SLCT         764A
@@STEPI        7674 @@TIME        74E5 @@VDCTL        7491
@@VER          7896 @@VRSEC       76C8 @@WEOF         78AB
@@WHERE        7386 @@WRITE       78CØ @@WRSEC        76F2
@@WRSSC        77Ø7 @@WRTRK       771C
26ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

MEMDISK/DCT - Memory disk driver

The Memdisk DCT program will establish a psuedo disk drive either in the main memory or in the alternate memory banks, if available. There must be room for Memdisk in the low driver zone or the installation will abort.

```
              ØØ1ØØ ;MEMDISK/ASM - Memory Disk Driver
ØØØØ          ØØ11Ø         TITLE   <MEMDISK/DCT - LS-DOS 6.2>
              ØØ12Ø ;
ØØØØ          ØØ13Ø *GET    SVCMAC:3                ;SVC Macro equivalents
              ØØØ1Ø ;SVCMAC/ASM - LS-DOS Version VI
              ØØØ2Ø *LIST   OFF
              Ø39ØØ *LIST   ON
ØØØØ          ØØ14Ø *GET    VALUES:3                ;Misc. equates
              Ø392Ø ;VALUES/ASM - Version 6
              Ø393Ø *LIST OFF
              Ø42ØØ *LIST ON
ØØØØ          ØØ15Ø *GET    COPYCOM:3               ;Copyright message
              Ø421Ø ; COPYCOM - File for Copyright COMment block
              Ø422Ø ;
ØØØØ          Ø423Ø         COM     '<*(C) 1982,83,84 by LSI*>'
              ØØ16Ø ;
ØAØØ          ØØ17Ø SDBPC   EQU     5*2*256         ;Single Density Bytes/Cyl
12ØØ          ØØ18Ø DDBPC   EQU     6*3*256         ;Double Density Bytes/Cyl
8ØØØ          ØØ19Ø LOWEST  EQU     8ØØØH           ;Lowest addr for Bank Ø
13ØØ          ØØ2ØØ HIDRVR  EQU     13ØØH           ;Highest addr for Driver
23ØØ          ØØ21Ø BUFFER$ EQU     23ØØH           ;Temporary I/O buffer Add
ØØØ3          ØØ22Ø MINCYL  EQU     3
ØØØF          ØØ23Ø WP      EQU     15              ;Write Prot Disk Error #
              ØØ24Ø ;
2CØØ          ØØ25Ø         ORG     2CØØH
              ØØ26Ø ;
              ØØ27Ø START
2CØØ          ØØ28Ø         @@CKBRKC                ;Check for break
2CØØ 3E6A     ØØØØ1         LD      A,1Ø6
2CØ2 EF       ØØØØ2         RST     4Ø
2CØ3 28Ø4     ØØ29Ø         JR      Z,STARTA        ;Continue if not
2CØ5 21FFFF   ØØ3ØØ         LD      HL,-1           ;  else abort
2CØ8 C9       ØØ31Ø         RET
              ØØ32Ø ;
2CØ9          ØØ33Ø STARTA  EQU     $
2CØ9 ED73222C ØØ34Ø         LD      (EXIT+1),SP     ;Save SP location
              ØØ35Ø ;
              ØØ36Ø ;       Install or Disable MemDISK
              ØØ37Ø ;
2CØD CDF62C   ØØ38Ø         CALL    CALCDRV         ;Calculate drive #
2C1Ø CD3F3Ø   ØØ39Ø         CALL    DOMEM           ;Get type of memdisk
2C13 CD9A2E   ØØ4ØØ         CALL    INSTMEM         ;Install MemDISK
              ØØ41Ø ;
              ØØ42Ø ;       Exit - Clean stack, Set HL, Revector <BREAK>
              ØØ43Ø ;
2C16 21ØØØØ   ØØ44Ø NORMEX  LD      HL,Ø            ;Normal Exit - HL = Ø
2C19 18Ø6     ØØ45Ø         JR      EXIT            ;Get SP & RETurn
              ØØ46Ø ;
2C1B CD2C2D   ØØ47Ø ABORT   CALL    GETDUP          ;Get duplicate DCT
2C1E 21FFFF   ØØ48Ø         LD      HL,-1           ;Abort
              ØØ49Ø ;
2C21 31ØØØØ   ØØ5ØØ EXIT    LD      SP,$-$          ;P/u SP address
2C24          ØØ51Ø         @@CKBRKC                ;Clear break
2C24 3E6A     ØØØØ3         LD      A,1Ø6
2C26 EF       ØØØØ4         RST     4Ø
2C27 C9       ØØ52Ø         RET
              ØØ53Ø ;
2C28          ØØ54Ø *GET    MEMDISKB:3
              Ø424Ø ;MEMDISKB/ASM - Miscellaneous Subroutines
2C28          Ø425Ø         SUBTTL  '<MEMDISKB - Subroutines>'
```

MEMDISKB - Subroutines

```
2C28                      PAGE
                 ;
                 ;        SETBANK - Tell system which banks are used
                 ;
2C28 3EØØ        SETBANK  LD      A,$-$             ;P/u bank #
2C2A 4F                   LD      C,A               ;Xfer to C
2C2B FEØ3                 CP      3                 ;Both banks 1 & 2 ?
2C2D 2ØØ5                 JR      NZ,STBANK         ;No - just 1 bank
2C2F ØD                   DEC     C                 ;Set C = 2
2C3Ø CD342C               CALL    STBANK            ;Show Bank in use
2C33 ØD                   DEC     C                 ;C = 1
2C34 C5          STBANK   PUSH    BC                ;Save BC
2C35 Ø6Ø3                 LD      B,3               ;Show in use function #
2C37                      @@BANK                    ;Let system know it
2C37 3E66                 LD      A,1Ø2
2C39 EF                   RST     4Ø
2C3A C1                   POP     BC
2C3B C9                   RET                       ;RETurn
                 ;
                 ;        FREBANK - Free up Bank C
                 ;
2C3C C5          FREBANK  PUSH    BC                ;Save C & B
2C3D Ø6Ø1                 LD      B,1               ;Show bank available
2C3F                      @@BANK
2C3F 3E66                 LD      A,1Ø2
2C41 EF                   RST     4Ø
2C42 C1                   POP     BC                ;Recover C
2C43 C9                   RET                       ;RETurn
                 ;
                 ;        DECASC2 - Display Number to video
                 ;
2C44 CDAE2C      DECASC2  CALL    SAVEREG           ;Save Registers
2C47 F5                   PUSH    AF                ;Save #
2C48 ØEØ8                 LD      C,BS              ;Backspace
2C4A CD592C               CALL    DSP               ;Output byte
2C4D CD592C               CALL    DSP               ;Twice
2C5Ø F1                   POP     AF                ;Recover A
2C51 CD632C               CALL    DECASC            ;Convert to ASCII
2C54 4C                   LD      C,H               ;P/u ms digit
2C55 CD592C               CALL    DSP
2C58 4D                   LD      C,L               ;P/u ls digit
                 ;
                 ;        DSP - Output byte to Video & exit if I/O err
                 ;
2C59             DSP      @@DSP                     ;Output byte
2C59 3EØ2                 LD      A,2
2C5B EF                   RST     4Ø
2C5C C8                   RET     Z                 ;RETurn if good
                 ;
                 ;        IOERR - Set HL = Error # & Abort
                 ;
2C5D 6F          IOERR    LD      L,A               ;Set HL = I/O Error #
2C5E 26ØØ                 LD      H,Ø
2C6Ø C3212C               JP      EXIT              ;Go to exit routine
                 ;
                 ;        Display Decimal ASCII equivalent
                 ;
2C63 262F        DECASC   LD      H,2FH             ;H=msb  of BCD ASCII
2C65 24          LPADD    INC     H                 ;Bump msb
2C66 D6ØA                 SUB     1Ø                ;Successive sub's of 1Ø
```

MEMDISKB - Subroutines

```
2C68 3ØFB               JR      NC,LPADD         ;Keep sub til carry
2C6A C63A               ADD     A,3AH            ;A = lsb ASCII
2C6C 6F                 LD      L,A              ;HL => DEC ASCII
2C6D C9                 RET
                ;
                ;       DECHEX - Convert Decimal ASCII to Hex
                ;
2C6E CD822C     DECHEX  CALL    GETDIG           ;Get digit
2C71 23                 INC     HL               ;Next byte in buffer
2C72 Ø5                 DEC     B                ;Dec digit counter
2C73 28ØB               JR      Z,DONE1          ;All done
2C75 57                 LD      D,A              ;Xfer to D
2C76 CD822C             CALL    GETDIG           ;Get digit
2C79 5F                 LD      E,A              ;Save digit
2C7A 7A                 LD      A,D              ;P/u ten's digit
2C7B 87                 ADD     A,A              ;Multiply
2C7C 87                 ADD     A,A              ;  A times 1Ø
2C7D 82                 ADD     A,D              ;  and add it
2C7E 87                 ADD     A,A              ;  to the ones digit
2C7F 83                 ADD     A,E              ;A = number of tracks
2C8Ø BF         DONE1   CP      A                ;Set Z flag
2C81 C9                 RET                      ; and RETurn
                ;
2C82 7E         GETDIG  LD      A,(HL)           ;P/u second digit
2C83 D63Ø               SUB     'Ø'              ;Cvt to binary
2C85 38Ø3               JR      C,ILLEGAL        ;Clr stack & RETurn NZ
2C87 FEØA               CP      1Ø               ;Legal digit
2C89 D8                 RET     C                ;Yes - A = digit
2C8A 3C         ILLEGAL INC     A                ;Reset Z flag
2C8B E1                 POP     HL               ;Clear stack
2C8C C9                 RET                      ;  and RETurn
                ;
                ;       Verify Error - P/u Bank / Address & display
                ;
2C8D E5         ERROR   PUSH    HL               ;L = lsb of Address
2C8E 3EC9               LD      A,ØC9H           ;Modify GETADR routine
2C9Ø 32722E             LD      (STFRET),A       ;HL <= page from DE
2C93 CD5D2E             CALL    GETADR
2C96 D1                 POP     DE               ;E = lsb of address
2C97 6B                 LD      L,E              ;HL = Bad RAM address
                ;
                ;       Stuff Bank # and Address into string
                ;
2C98 3E3Ø               LD      A,'Ø'            ;Cvt BANK # to ASCII
2C9A 81                 ADD     A,C
2C9B 324737             LD      (VBANK),A        ;Stuff into string
2C9E EB                 EX      DE,HL            ;Xfer address to DE
2C9F 215737             LD      HL,VLOC          ;HL => string destination
2CA2                    @@HEX16                  ;Cvt DE to Hex ASCII @ HL
2CA2 3E63               LD      A,99
2CA4 EF                 RST     4Ø
                ;
                ;       Display string & restore hi/low mem
                ;
2CA5 213137             LD      HL,BADRAM        ;"BAD RAM ...
2CA8                    @@LOGOT                  ;Display it
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
```

MEMDISKB - Subroutines

```
2CA8 3EØC      ØØØ16          LD      A,12
2CAA EF        ØØØ17          RST     4Ø
2CAB C3EE2E    Ø534Ø          JP      OLDRVR            ;Leave & clear stack
               Ø535Ø ;
               Ø536Ø ;        SAVEREG - Save All Primary Registers
               Ø537Ø ;
2CAE E3        Ø538Ø SAVEREG  EX      (SP),HL
2CAF 22C42C    Ø539Ø          LD      (RETADDR+1),HL
2CB2 E1        Ø54ØØ          POP     HL
2CB3 E5        Ø541Ø          PUSH    HL
2CB4 ED53B832  Ø542Ø          LD      (SAVEDE),DE
2CB8 D5        Ø543Ø          PUSH    DE
2CB9 C5        Ø544Ø          PUSH    BC
2CBA F5        Ø545Ø          PUSH    AF
2CBB 11C62C    Ø546Ø          LD      DE,RESTREG
2CBE D5        Ø547Ø          PUSH    DE
2CBF ED5BB832  Ø548Ø          LD      DE,(SAVEDE)
2CC3 C3ØØØØ    Ø549Ø RETADDR  JP      $-$
2CC6 F1        Ø55ØØ RESTREG  POP     AF
2CC7 C1        Ø551Ø          POP     BC
2CC8 D1        Ø552Ø          POP     DE
2CC9 E1        Ø553Ø          POP     HL
2CCA C9        Ø554Ø          RET
               Ø555Ø ;
               Ø556Ø ;        CKBANK - Check if Bank C is in use
               Ø557Ø ;
2CCB C5        Ø558Ø CKBANK   PUSH    BC                ;Save BC
2CCC Ø6Ø2      Ø559Ø          LD      B,2               ;Bank in use ?
2CCE           Ø56ØØ          @@BANK                    ;Check it out
2CCE 3E66      ØØØ18          LD      A,1Ø2
2CDØ EF        ØØØ19          RST     4Ø
2CD1 C1        Ø561Ø          POP     BC                ;Recover BC
2CD2 C8        Ø562Ø          RET     Z                 ;RETurn if available
2CD3 C3DA32    Ø563Ø          JP      BNKUSE            ;  else - display "in use"
               Ø564Ø ;
               Ø565Ø ;        INPUT - Input a line to the input buffer
               Ø566Ø ;
2CD6 21ØØ39    Ø567Ø INPUT    LD      HL,BUFFER         ;HL => Input buffer
2CD9           Ø568Ø          @@KEYIN                   ;Input line
2CD9 3EØ9      ØØØ2Ø          LD      A,9
2CDB EF        ØØØ21          RST     4Ø
2CDC DA1B2C    Ø569Ø          JP      C,ABORT           ;Exit if <BREAK> hit
2CDF Ø4        Ø57ØØ          INC     B                 ;Set Z if no chars
2CEØ Ø5        Ø571Ø          DEC     B
2CE1 C9        Ø572Ø          RET                       ;  else RETurn
               Ø573Ø ;
               Ø574Ø ;        GETCYL - Get max # of cylinders in A
               Ø575Ø ;
2CE2 D5        Ø576Ø GETCYL   PUSH    DE                ;Save regs
2CE3 E5        Ø577Ø          PUSH    HL
               Ø578Ø ;
               Ø579Ø ;        Init DE = # bytes/cyl, A = dividend (-1)
               Ø58ØØ ;
2CE4 11ØØ12    Ø581Ø BPC      LD      DE,DDBPC          ;P/u bytes/cyl
2CE7 3EFF      Ø582Ø          LD      A,-1              ;Init avail cyl cnt = -1
               Ø583Ø ;
               Ø584Ø ;        Divide total bytes available by Bytes/cyl
               Ø585Ø ;
2CE9 3C        Ø586Ø DIVLP    INC     A                 ;Bump cyl count
```

```
2CEA B7                       OR      A
2CEB ED52                     SBC     HL,DE             ;Take off 1 cyl
2CED 30FA                     JR      NC,DIVLP          ;Loop until carry
                    ;
                    ;         A = # of cyls avail, Restore regs
                    ;
2CEF E1                       POP     HL                ;Recover regs
2CF0 D1                       POP     DE
                    ;
                    ;         Set Z flag if more than 1 cylinder available
                    ;
2CF1 FE02                     CP      2                 ;0 or 1 ?
2CF3 D8                       RET     C                 ;Yes - RETurn NZ
2CF4 BF                       CP      A                 ;Set Z flag
2CF5 C9                       RET                       ;  and RETurn
                    ;
                    ;         CALCDRV - Calculate drive Number for MemDISK
                    ;
                    ;         DE => DCT block for Drive
                    ;
2CF6                CALCDRV   EQU     $
2CF6 EB                       EX      DE,HL             ;Xfer to HL
2CF7 22BA32                   LD      (SAVEDCT),HL      ;Save DCT pointer
2CFA CD202D                   CALL    SAVDCT            ;Save DCT
2CFD 7C                       LD      A,H               ;Drive number issued ?
2CFE B5                       OR      L
2CFF CABE32                   JP      Z,NODRV           ;No drive entered
                    ;
                    ;         Get Start of Drive Code Table
                    ;
2D02 0E00                     LD      C,0               ;Get start of DCT
2D04                          @@GTDCT                   ;Get DCT for Drive 0
2D04 3E51                     LD      A,81
2D06 EF                       RST     40
2D07 FDE5                     PUSH    IY                ;Get DCT start
2D09 D1                       POP     DE
                    ;
                    ;         Calculate Offset in Table
                    ;
2D0A AF                       XOR     A
2D0B ED52                     SBC     HL,DE             ;L = offset from start
2D0D B5                       OR      L                 ;P/u offset
2D0E CAC232                   JP      Z,BADDRV          ;Cannot use DRIVE 0
                    ;
                    ;         Divide offset by 10 to get drive #
                    ;
2D11 06FF                     LD      B,-1              ;Init dividend = -1
2D13 04             DIVLP1    INC     B                 ;Bump dividend
2D14 D60A                     SUB     10                ;Subtract ten
2D16 30FB                     JR      NC,DIVLP1
                    ;
                    ;         Stuff away drive # into WRSEC routine
                    ;
2D18 78                       LD      A,B               ;P/u drive #
2D19 32C52F                   LD      (DRIVE+1),A       ;Stuff away drive #
                    ;
                    ;         Point IY to System Flag table & RETurn
                    ;
2D1C                          @@FLAGS                   ;IY => Flags
```

MEMDISKB - Subroutines

```
2D1C 3E65      ØØØ24          LD      A,1Ø1
2D1E EF        ØØØ25          RST     4Ø
2D1F C9        Ø644Ø          RET                         ;Later
               Ø645Ø ;
               Ø646Ø ;        SAVDCT - Save Old DCT setup
               Ø647Ø ;
2D2Ø CDAE2C    Ø648Ø SAVDCT   CALL    SAVEREG             ;Save registers
2D23 11ØØ3A    Ø649Ø          LD      DE,DUPDCT           ;Destination
2D26 Ø1ØAØØ    Ø65ØØ DOXFER1  LD      BC,1Ø               ;1Ø bytes to xfer
2D29 EDBØ      Ø651Ø          LDIR
2D2B C9        Ø652Ø          RET
               Ø653Ø ;
               Ø654Ø ;        GETDUP - Get Duplicate of original DCT setup
               Ø655Ø ;
2D2C ED5BBA32  Ø656Ø GETDUP   LD      DE,(SAVEDCT)        ;DE => DCT+Ø
2D3Ø 21ØØ3A    Ø657Ø          LD      HL,DUPDCT           ;Source
2D33 18F1      Ø658Ø          JR      DOXFER1             ;Transfer back
               Ø659Ø ;
               Ø66ØØ ;        GTDRV - P/u Next available Driver Address
               Ø661Ø ;
               Ø662Ø ;        IX <= Driver Address Pointer
               Ø663Ø ;        DE <= Current Address
               Ø664Ø ;
2D35 E5        Ø665Ø GTDRV    PUSH    HL                  ;Save HL
2D36 114B49    Ø666Ø          LD      DE,'IK'             ;P/u *KI DCB address
2D39           Ø667Ø          @@GTDCB
2D39 3E52      ØØØ26          LD      A,82
2D3B EF        ØØØ27          RST     4Ø
2D3C 2B        Ø668Ø          DEC     HL                  ;KIDCB - 2 => free area
2D3D E5        Ø669Ø          PUSH    HL                  ;Xfer to IX
2D3E DDE1      Ø67ØØ          POP     IX
2D4Ø 56        Ø671Ø          LD      D,(HL)              ;P/u address in DE
2D41 2B        Ø672Ø          DEC     HL
2D42 22F82E    Ø673Ø          LD      (KIDCB$+1),HL       ;Save address to stuff
2D45 5E        Ø674Ø          LD      E,(HL)
2D46 E1        Ø675Ø          POP     HL                  ;Recover HL
2D47 C9        Ø676Ø          RET
               Ø677Ø ;
               Ø678Ø ;        INSTDRV - Relocate & Install Disk Driver
               Ø679Ø ;
2D48 EB        Ø68ØØ INSTDRV  EX      DE,HL               ;Xfer dest to HL
2D49 11BE2D    Ø681Ø          LD      DE,DRIVER           ;Start of driver
2D4C E5        Ø682Ø          PUSH    HL                  ;Save Source & Dest ptrs
2D4D D5        Ø683Ø          PUSH    DE
2D4E B7        Ø684Ø          OR      A                   ;Clear carry
2D4F ED52      Ø685Ø          SBC     HL,DE               ;Get offset
               Ø686Ø ;
               Ø687Ø ;        Relocate internal references in driver
               Ø688Ø ;
2D51 DD217Ø2D  Ø689Ø          LD      IX,RELTBL           ;Point to relocation tbl
2D55 44        Ø69ØØ          LD      B,H                 ;Move to BC
2D56 4D        Ø691Ø          LD      C,L
2D57 DD6EØØ    Ø692Ø RLOOP    LD      L,(IX)              ;Get address to change
2D5A DD66Ø1    Ø693Ø          LD      H,(IX+1)
2D5D 7C        Ø694Ø          LD      A,H
2D5E B5        Ø695Ø          OR      L
2D5F 2829      Ø696Ø          JR      Z,RELDUN
2D61 5E        Ø697Ø          LD      E,(HL)              ;P/U address
2D62 23        Ø698Ø          INC     HL
```

MEMDISKB - Subroutines

```
2D63 56         Ø699Ø         LD      D,(HL)
2D64 EB         Ø7ØØØ         EX      DE,HL          ;Offset it
2D65 Ø9         Ø7Ø1Ø         ADD     HL,BC
2D66 EB         Ø7Ø2Ø         EX      DE,HL
2D67 72         Ø7Ø3Ø         LD      (HL),D         ;Put it back
2D68 2B         Ø7Ø4Ø         DEC     HL
2D69 73         Ø7Ø5Ø         LD      (HL),E
2D6A DD23       Ø7Ø6Ø         INC     IX
2D6C DD23       Ø7Ø7Ø         INC     IX
2D6E 18E7       Ø7Ø8Ø         JR      RLOOP          ;Loop till done
                Ø7Ø9Ø ;
                Ø71ØØ ;       Relocation Table for Driver
                Ø711Ø ;
2D7Ø ØØ2E       Ø712Ø RELTBL  DW      REL1+1,REL2+1,REL3+1,REL4+1
     1Ø2E 412E 472E
2D78 7A2E       Ø713Ø         DW      REL5+1,REL6+2,REL7+1,REL8+1,REL8A+1
     E22D E62D EC2D 362E
2D82 7D2E       Ø714Ø         DW      REL8B+1,REL9+1,REL2A+1,Ø
     E92D Ø72E ØØØØ
                Ø715Ø ;
                Ø716Ø ;       Transfer MemDisk driver to driver area
                Ø717Ø ;
2D8A E1         Ø718Ø RELDUN  POP     HL             ;HL => Source DE => Dest
2D8B D1         Ø719Ø         POP     DE
2D8C D5         Ø72ØØ         PUSH    DE             ;Save start
2D8D Ø1DCØØ     Ø721Ø         LD      BC,LENGTH      ;# bytes to move
2D9Ø EDBØ       Ø722Ø         LDIR                   ;Block move
2D92 D1         Ø723Ø         POP     DE             ;Restore start
2D93 C9         Ø724Ø         RET                    ;RETurn
                Ø725Ø ;
                Ø726Ø ;       SETDCT - Set up Drive Code Table for MemDISK
                Ø727Ø ;
2D94 DD2ABA32   Ø728Ø SETDCT  LD      IX,(SAVEDCT)   ;IX => DCT address
2D98 DD36ØØC3   Ø729Ø         LD      (IX+Ø),ØC3H    ;Enable
2D9C DD73Ø1     Ø73ØØ         LD      (IX+1),E       ;Lsb of driver
2D9F DD72Ø2     Ø731Ø         LD      (IX+2),D       ;Msb of driver
2DA2 DD36Ø34Ø   Ø732Ø SDEND   LD      (IX+3),4ØH     ;DD,5",floppy,step=6
2DA6 DD36Ø45Ø   Ø733Ø SDENE   LD      (IX+4),5ØH     ;DDC=Y, 1 side, ALIEN
2DAA DD36Ø5ØØ   Ø734Ø         LD      (IX+5),Ø       ;Current Cyl = Ø
2DAE DD77Ø6     Ø735Ø         LD      (IX+6),A       ;# of tracks rel from Ø
2DB1 DD36Ø711   Ø736Ø SDENF   LD      (IX+7),17      ;18 spt (DD), 1Ø spt (SD)
2DB5 DD36Ø845   Ø737Ø SDENG   LD      (IX+8),45H     ;2/3 G/C, 5/6 S/G
2DB9 DD36Ø9Ø1   Ø738Ø         LD      (IX+9),1       ;Directory Cyl = 1
2DBD C9         Ø739Ø         RET                    ;RETurn
2DBE            ØØ55Ø *GET    MEMDISKC:3
                Ø74ØØ ;MEMDISKC/ASM - MemDISK Driver Code
2DBE            Ø741Ø         SUBTTL  '<MEMDISKC - MemDISK Driver>'
```

MEMDISKC - MemDISK Driver

```
2DBE                      PAGE
                  ;
2DBE 181D         DRIVER  JR      INIT            ;Jump around header
2DCØ ØØØØ         OLDHIGH DW      Ø               ;Old HIDRV$
2DC2 Ø3                   DB      3,'$MD'         ;Header
     24 4D 44
2DC6 ØØØØ         OLD_HI  DW      Ø               ;Old HIGH$ (for bank Ø)
2DC8 ØØ           BANKIM  DB      ØØØØØØØØB       ;Bank Image
2DC9 ØØØØ         DRVLOW  DW      Ø               ;What driver addr was
2DCB ØØØØ         MEMHIGH DW      Ø               ;HIGH$ after installed
                  ;
                          IF      @MOD2
                          DC      32,Ø            ;Model 2 stack area
                  ;
                          ELSE
2DCD ØØ                   DC      16,Ø            ;Driver Stack Area
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
                          ENDIF
2DDD              MYSTACK EQU     $               ;Start of Mystack
                  ;
                  ;       Reset SP to MYSTACK, and CALL driver
                  ;
2DDD E5           INIT    PUSH    HL              ;Save Registers
2DDE D5                   PUSH    DE              ;
2DDF C5                   PUSH    BC              ;
2DEØ ED73EF2D     REL6    LD      (SAVESP+1),SP   ;Save original SP
2DE4 F3                   DI                      ;Interrupts off
2DE5 31DD2D       REL7    LD      SP,MYSTACK      ;Memdisk SP
2DE8 228C2E       REL9    LD      (BUFF+1),HL     ;Save buffer addr request
2DEB CDF62D       REL8    CALL    MEMDRIV         ;Call the actual driver
2DEE 31ØØØØ       SAVESP  LD      SP,$-$          ;P/u original SP
2DF1 FB                   EI                      ;Back on
2DF2 C1                   POP     BC              ;Restore Registers
2DF3 D1                   POP     DE
2DF4 E1                   POP     HL
2DF5 C9                   RET
                  ;
2DF6 78           MEMDRIV LD      A,B             ;Get operation byte
                  ;
2DF7 FEØ9         B9      CP      9               ;Operation #9 ?
2DF9 2Ø27                 JR      NZ,B1Ø          ;No - Check for Verify
                  ;
                  ;       READ sector - Set Z if D = directory cyl
                  ;
2DFB 15                   DEC     D               ;Set Z flag if Cyl = 1
2DFC F5                   PUSH    AF
2DFD 14                   INC     D               ;Restore cyl #
                  ;
                  ;       Set up For transfer to temporary I/O buffer
                  ;
2DFE E5                   PUSH    HL              ;Save User I/O buffer ptr
2DFF CD5D2E       REL1    CALL    GETADR          ;HL => MemDISK Sector
2EØ2 38Ø8                 JR      C,DOXFER        ;High - use temporary buf
                  ;
                  ;       I/O buff is low - xfer MemDISK sector to it
                  ;
2EØ4 EDBØ                 LDIR                    ;Xfer directly to buffer
2EØ6 CD832E       REL2A   CALL    GETOLD          ;Get original bank
2EØ9 E1                   POP     HL              ;HL => User I/O buffer
```

MEMDISKC - MemDISK Driver

```
2EØA 18ØD      Ø799Ø          JR      CHKDIR2          ;Check if directory cyl
               Ø8ØØØ ;
               Ø8Ø1Ø ;        Transfer MemDISK sector to Temporary Buffer
               Ø8Ø2Ø ;
2EØC D5        Ø8Ø3Ø DOXFER   PUSH    DE               ;DE => Temporary Buffer
2EØD EDBØ      Ø8Ø4Ø          LDIR                     ;Xfer to system area
               Ø8Ø5Ø ;
               Ø8Ø6Ø ;        Xfer data from temporary to User Buffer
               Ø8Ø7Ø ;
2EØF CD832E    Ø8Ø8Ø REL2     CALL    GETOLD           ;Get original bank
2E12 E1        Ø8Ø9Ø          POP     HL               ;HL => Temporary buffer
2E13 D1        Ø81ØØ          POP     DE               ;DE => User I/O buffer
2E14 Ø1ØØØ1    Ø811Ø          LD      BC,256           ;BC = 256 bytes to xfer
2E17 EDBØ      Ø812Ø          LDIR                     ;Xfer to user buffer
               Ø813Ø ;
               Ø814Ø ;        Set A = Error #6 if Cylinder 1 (Directory)
               Ø815Ø ;
2E19 F1        Ø816Ø CHKDIR2  POP     AF               ;Get Z
2E1A 2ØØ4      Ø817Ø CHKDIR   JR      NZ,NOTDIR        ;Not a directory read
2E1C 3EØ6      Ø818Ø          LD      A,6              ;Error Code = 6
2E1E B7        Ø819Ø          OR      A                ;NZ condition
2E1F C9        Ø82ØØ          RET                      ;And RETurn
2E2Ø AF        Ø821Ø NOTDIR   XOR     A                ;Set Z flag
2E21 C9        Ø822Ø          RET                      ;And return
               Ø823Ø ;
2E22 FEØA      Ø824Ø B1Ø      CP      1Ø               ;Verify sector ?
2E24 2ØØ3      Ø825Ø          JR      NZ,B13           ;Check more if not
               Ø826Ø ;
               Ø827Ø ;        Verify a sector
               Ø828Ø ;
2E26 15        Ø829Ø          DEC     D                ;Directory Cylinder
2E27 18F1      Ø83ØØ          JR      CHKDIR           ;Check if Directory cyl
               Ø831Ø ;
2E29 FEØD      Ø832Ø B13      CP      13               ;Write a sector?
2E2B 2Ø1E      Ø833Ø          JR      NZ,B14           ;Check further if not
               Ø834Ø ;
               Ø835Ø ;        Write A Sector
               Ø836Ø ;
2E2D 3EØF      Ø837Ø WRITES   LD      A,WP             ;WP error X'ØF'
2E2F FDCBØ37E  Ø838Ø          BIT     7,(IY+3)         ;Software Write Protect?
2E33 CØ        Ø839Ø          RET     NZ               ;Return with error
               Ø84ØØ ;
               Ø841Ø ;        Set up for Tranfer to Temporary Buffer
               Ø842Ø ;
2E34 D5        Ø843Ø          PUSH    DE               ;Save Cyl/Sector
2E35 CD8A2E    Ø844Ø REL8A    CALL    GETBUF           ;Get buffer ptr
2E38 3ØØ5      Ø845Ø          JR      NC,RECVDE        ;Get back DE
2E3A Ø1ØØØ1    Ø846Ø          LD      BC,256           ;BC = 256 bytes to xfer
2E3D EDBØ      Ø847Ø          LDIR                     ;Xfer to temp buffer
2E3F D1        Ø848Ø RECVDE   POP     DE               ;DE = Cyl/sector
               Ø849Ø ;
               Ø85ØØ ;        Get Sector from MemDISK & xfer to User buff
               Ø851Ø ;
2E4Ø CD5D2E    Ø852Ø REL3     CALL    GETADR           ;HL <= Mem, DE <= Buffer
2E43 EB        Ø853Ø          EX      DE,HL
2E44 EDBØ      Ø854Ø          LDIR                     ;Xfer to user buffer
2E46 CD832E    Ø855Ø REL4     CALL    GETOLD           ;Get original back
2E49 AF        Ø856Ø          XOR     A                ;Set Z flag
2E4A C9        Ø857Ø          RET
```

MEMDISKC - MemDISK Driver

```
                Ø858Ø ;
2E4B FEØE       Ø859Ø B14     CP      14              ;Write system sector?
2E4D 28DE       Ø86ØØ         JR      Z,WRITES        ;Go if so
                Ø861Ø ;
2E4F FEØC       Ø862Ø         CP      12              ;Format command?
2E51 28Ø4       Ø863Ø         JR      Z,B14A          ;Go if so
2E53 FEØF       Ø864Ø         CP      15              ;Write Track ?
2E55 2ØØ4       Ø865Ø         JR      NZ,EX1          ;No - exit Z
2E57 3EØ8       Ø866Ø B14A    LD      A,8             ;Yes - Exit NZ
2E59 B7         Ø867Ø         OR      A               ;Error = Device not avail
2E5A C9         Ø868Ø         RET
                Ø869Ø ;
2E5B AF         Ø87ØØ EX1     XOR     A               ;Zero A, set Z
2E5C C9         Ø871Ø         RET                     ;Return with Z set
                Ø872Ø ;
                Ø873Ø ;       GETADR - Point HL to MemDISK area
                Ø874Ø ;        - Point DE to Temporary buffer
                Ø875Ø ;        - Set BC = 256 (bytes to xfer)
                Ø876Ø ;
2E5D 7A         Ø877Ø GETADR  LD      A,D             ;P/u Cylinder #
                Ø878Ø ;
                Ø879Ø ;       Multiply cylinder # x 1Ø or 18 (sectors/cyl)
                Ø88ØØ ;
2E5E 87         Ø881Ø SDENA   ADD     A,A             ;X 2 or NOP if Single Den
2E5F 57         Ø882Ø         LD      D,A             ;DDEN = x 2  SDEN = x 1
2E6Ø 87         Ø883Ø         ADD     A,A             ;DDEN = x 4  SDEN = x 2
2E61 87         Ø884Ø         ADD     A,A             ;DDEN = x 8  SDEN = x 4
2E62 87         Ø885Ø SDENB   ADD     A,A             ;DDEN = x 16  SDEN = x 5
2E63 82         Ø886Ø SDENC   ADD     A,D             ;DDEN = x 18  SDEN = x 1Ø
                Ø887Ø ;
                Ø888Ø ;       Add Sect offset (E) & add 8ØH if bank 2 active
                Ø889Ø ;
2E64 83         Ø89ØØ         ADD     A,E             ;Add sector offset
2E65 C6ØØ       Ø891Ø OFFSET  ADD     A,$-$           ;8ØH if 2 active
                Ø892Ø ;
                Ø893Ø ;       Set HL => sector, C = Default bank (Ø or 1)
                Ø894Ø ;
2E67 67         Ø895Ø         LD      H,A             ;Stuff msb in H
2E68 2EØØ       Ø896Ø         LD      L,Ø             ;Land on page boundary
2E6A ØEØØ       Ø897Ø DEFBANK LD      C,$-$           ;C = Ø or C = 1
                Ø898Ø ;
                Ø899Ø ;       Set C = Bank #2 if Address > X'7FFF'
                Ø9ØØØ ;
2E6C Ø7         Ø9Ø1Ø         RLCA                    ;Address > X'7FFF' ?
2E6D 3ØØ1       Ø9Ø2Ø         JR      NC,GOTBANK      ;No - got it
2E6F ØC         Ø9Ø3Ø         INC     C               ;Yes - Set C = 2
                Ø9Ø4Ø ;
                Ø9Ø5Ø ;       Force address > X'7FFF' & Select Bank C
                Ø9Ø6Ø ;
2E7Ø CBFC       Ø9Ø7Ø GOTBANK SET     7,H             ;Force Address > X'7FFF'
2E72 45         Ø9Ø8Ø STFRET  LD      B,L             ;Bring in Bank C
2E73            Ø9Ø9Ø         @@BANK
2E73 3E66       ØØØ28         LD      A,1Ø2
2E75 EF         ØØØ29         RST     4Ø
                Ø91ØØ ;
                Ø911Ø ;       Pick up Bank previously in use & Save
                Ø912Ø ;
2E76 79         Ø913Ø         LD      A,C             ;P/u last bank
2E77 E67F       Ø914Ø         AND     7FH             ;Ignore Hi-bit
```

MEMDISKC - MemDISK Driver

```
2E79 32842E     Ø915Ø REL5     LD       (GETOLD+1),A     ;  and stuff away
                Ø916Ø ;
                Ø917Ø ;        Set DE => Overlay Buffer, BC = 256
                Ø918Ø ;
2E7C CD8A2E     Ø919Ø REL8B    CALL     GETBUF           ;Get buffer ptr
2E7F Ø1ØØØ1     Ø92ØØ          LD       BC,256           ;Set BC = 256
2E82 C9         Ø921Ø          RET
                Ø922Ø ;
                Ø923Ø ;        OLDBNK - Get original Bank used
                Ø924Ø ;
2E83 Ø1ØØØØ     Ø925Ø GETOLD   LD       BC,$-$           ;B = Ø, C = Bank #
2E86            Ø926Ø          @@BANK                    ;Get bank
2E86 3E66       ØØØ3Ø          LD       A,1Ø2
2E88 EF         ØØØ31          RST      4Ø
2E89 C9         Ø927Ø          RET
                Ø928Ø ;
                Ø929Ø ;        GETBUF - Get Buffer ptr to LDIR from or to
                Ø93ØØ ;
2E8A E5         Ø931Ø GETBUF   PUSH     HL               ;Save source/dest ptr
2E8B 11ØØØØ     Ø932Ø BUFF     LD       DE,$-$           ;P/u requested I/O buffer
2E8E 21ØØ7F     Ø933Ø          LD       HL,7FØØH         ;Use (BUFF+1) if < 7FØØH
2E91 B7         Ø934Ø          OR       A
2E92 ED52       Ø935Ø          SBC      HL,DE            ;Past 7FØØH ?
2E94 E1         Ø936Ø          POP      HL               ;Rcvr ptr
2E95 DØ         Ø937Ø          RET      NC               ;No - use requested buff
2E96 11ØØ23     Ø938Ø          LD       DE,BUFFER$       ;Yes - use BUFFER$
2E99 C9         Ø939Ø          RET
                Ø94ØØ ;
ØØDC            Ø941Ø LENGTH   EQU      $-DRIVER         ;Length of Driver
2E9A            ØØ56Ø *GET     MEMDISKA:3
                Ø942Ø ;MEMDISKA/ASM - Memdisk Initialization
2E9A            Ø943Ø          SUBTTL   '<MEMDISKA - Installation>'
```

MEMDISKA - Installation

```
2E9A            Ø944Ø           PAGE
                Ø945Ø ;
2E9A F5         Ø946Ø INSTMEM PUSH    AF                ;Save # cyls
2E9B C5         Ø947Ø         PUSH    BC                ;Save Bank #
                Ø948Ø ;
                Ø949Ø ;       Is there a MemDISK driver trapped ?
                Ø95ØØ ;
2E9C 11CØ34     Ø951Ø         LD      DE,MD$            ;"$MD"
2E9F            Ø952Ø         @@GTMOD                   ;MemDISK in ?
2E9F 3E53       ØØØ32         LD      A,83
2EA1 EF         ØØØ33         RST     4Ø
2EA2 2Ø11       Ø953Ø         JR      NZ,NOT__IN        ;No
                Ø954Ø ;
                Ø955Ø ;       There is a driver trapped - use that area
                Ø956Ø ;
2EA4 22EF2E     Ø957Ø         LD      (OLDRVR+1),HL     ;Save old driver addr
2EA7 EB         Ø958Ø         EX      DE,HL             ;Pt DE => Destination
2EA8 216737     Ø959Ø         LD      HL,RE_USE         ;Set re-use flag
2EAB 34         Ø96ØØ         INC     (HL)
2EAC 21DBØØ     Ø961Ø         LD      HL,LENGTH-1       ;Set HL = last used
2EAF 19         Ø962Ø         ADD     HL,DE             ;  address of driver
2EBØ 22CØ2D     Ø963Ø         LD      (OLDHIGH),HL      ;Xfer into driver
2EB3 1827       Ø964Ø         JR      DO_INST           ;Install driver
                Ø965Ø ;
                Ø966Ø ;       Driver is not in memory - is there room ?
                Ø967Ø ;
2EB5 CD352D     Ø968Ø NOT__IN CALL    GTDRV             ;P/u low driver ptr
2EB8 ED53EF2E   Ø969Ø         LD      (OLDRVR+1),DE     ;Save it
2EBC 21DBØØ     Ø97ØØ         LD      HL,LENGTH-1       ;HL = length of driver
2EBF Ø1ØØ13     Ø971Ø         LD      BC,HIDRVR         ;BC = 1 + highest avail
2EC2 19         Ø972Ø         ADD     HL,DE             ;HL => Last used by Mem
2EC3 22CØ2D     Ø973Ø         LD      (OLDHIGH),HL
2EC6 23         Ø974Ø         INC     HL
2EC7 B7         Ø975Ø         OR      A
2EC8 E5         Ø976Ø         PUSH    HL                ;Will MemDisk fit ?
2EC9 ED42       Ø977Ø         SBC     HL,BC
2ECB E1         Ø978Ø         POP     HL
2ECC 38Ø8       Ø979Ø         JR      C,OKTOGO          ;Yes - let's do it
                Ø98ØØ ;
                Ø981Ø ;       Insufficient Driver space
                Ø982Ø ;
2ECE 21D232     Ø983Ø         LD      HL,NOMEM          ;Alter exit message
2ED1 22112F     Ø984Ø         LD      ($NOT+1),HL
2ED4 1818       Ø985Ø         JR      OLDRVR            ;Reclaim hi mem if bank Ø
                Ø986Ø ;
                Ø987Ø ;       Save next avail mem addr & set Memdisk bit
                Ø988Ø ;
2ED6 DD74ØØ     Ø989Ø OKTOGO  LD      (IX),H            ;Stuff msb
2ED9 DD75FF     Ø99ØØ         LD      (IX-1),L          ;Stuff lsb
                Ø991Ø ;
                Ø992Ø ;       Install MemDISK driver & set up DCT
                Ø993Ø ;
2EDC CD482D     Ø994Ø DO_INST CALL    INSTDRV           ;Relocate, install driver
2EDF C1         Ø995Ø         POP     BC                ;C = Bank # requests
2EEØ F1         Ø996Ø         POP     AF                ;A = # cylinders
2EE1 CD942D     Ø997Ø         CALL    SETDCT            ;Set up DCT
                Ø998Ø ;
                Ø999Ø ;       Prompt for Format
                1ØØØØ ;
2EE4 CD7C32     1ØØ1Ø         CALL    FORMTIT           ;Format this ?
```

MEMDISKA - Installation

```
2EE7 282A      10020          JR      Z,DOFORM1         ;Yes - do it
               10030 ;
               10040 ;        Format = No, Is there a MemDISK here ?
               10050 ;
2EE9 3E00      10060 MEMIN1   LD      A,$-$             ;0 = not active
2EEB B7        10070          OR      A                 ;
2EEC 2034      10080          JR      NZ,SHOWINU        ;MemDisk previously in
               10090 ;
               10100 ;        Abort installation - stuff X'C9' in DCT
               10110 ;
2EEE 210000    10120 OLDRVR   LD      HL,$-$            ;P/u original driver addr
2EF1 3A6737    10130          LD      A,(RE_USE)        ;Have we re-used driver
2EF4 B7        10140          OR      A                 ;  area that was trapped ?
2EF5 2003      10150          JR      NZ,DONTRES        ;Yes - don't reset memptr
2EF7 220000    10160 KIDCB$   LD      ($-$),HL          ;Stuff ptr used
2EFA 2ABA32    10170 DONTRES  LD      HL,(SAVEDCT)      ;P/u DCT address
2EFD 36C9      10180          LD      (HL),0C9H         ;Disable it
2EFF FDCB03A6  10190          RES     4,(IY+DFLAG$)     ;Reset MemDISK bit
2F03 3A292C    10200          LD      A,(SETBANK+1)     ;P/u bank request
2F06 B7        10210          OR      A                 ;If alternate bank(s),
2F07 2007      10220          JR      NZ,$NOT           ;  don't reset high$
2F09 2AC634    10230          LD      HL,(MDDATA+2)     ;Pu old high$
2F0C 47        10240          LD      B,A
2F0D           10250          @@HIGH$                   ;Reset high$
2F0D 3E64      00034          LD      A,100
2F0F EF        00035          RST     40
2F10 C3DE32    10260 $NOT     JP      NOTACT            ;Show not installed
               10270 ;
               10280 ;        Format mem, init GAT & HIT, & BOOT-DIR entries
               10290 ;
2F13 CD2D32    10300 DOFORM1  CALL    FORMAT            ;Format
2F16 CD302F    10310          CALL    WRBOOT            ;Write BOOT/SYS
2F19 CD5A2F    10320          CALL    WRGAT             ;Initialize GAT
2F1C CDD52F    10330          CALL    WRHIT             ;Initialize HIT
2F1F CDE22F    10340          CALL    WRENT             ;Put DIR & BOOT entries
2F22 CD282C    10350 SHOWINU  CALL    SETBANK           ;Show Banks in use
2F25 FDCB03E6  10360          SET     4,(IY+DFLAG$)     ;Set MemDisk flag
2F29 211335    10370          LD      HL,INSTALD        ;Init"MemDisk Installed
2F2C           10380          @@LOGOT                   ;Display the msg
               00036          IFEQ    00H,1
               00037          LD      HL,
               00038          ENDIF
2F2C 3E0C      00039          LD      A,12
2F2E EF        00040          RST     40
2F2F C9        10390          RET                       ;Done - GO TO EXIT
               10400 ;
               10410 ;        WRBOOT - Write BOOT/SYS information
               10420 ;
2F30 AF        10430 WRBOOT   XOR     A                 ;Fill byte
2F31 210038    10440          LD      HL,IOBUFF         ;HL => I/O buffer
               10450 ;
               10460 ;        Fill BOOT/SYS with Zeroes
               10470 ;
2F34 77        10480 FILBUF   LD      (HL),A            ;Stuff in byte
2F35 2C        10490          INC     L                 ;One sector to
2F36 20FC      10500          JR      NZ,FILBUF         ;  fill
               10510 ;
               10520 ;        Write # of Sectors in BOOT
               10530 ;
```

MEMDISKA - Installation

```
2F38 57                LD      D,A             ;Cylinder Ø
2F39 5F                LD      E,A             ;Sector Ø
2F3A Ø6Ø6     BTSECS   LD      B,6             ;P/u Sec cnt - 5,6, or 18
2F3C CDC12F   BTLP     CALL    WRSEC           ;Write sector
2F3F 1C                INC     E               ;Bump
2F4Ø 1ØFA              DJNZ    BTLP
              ;
              ;        Write Directory Cylinder byte in Sector Zero
              ;
2F42 2EØ2              LD      L,2             ;Byte 2
2F44 36Ø1              LD      (HL),1          ;Directory cyl = 1
              ;
              ;        Write Sector Ø of Cylinder Ø
              ;
2F46 11ØØØØ            LD      DE,Ø            ;Cylinder Ø, Sector Ø
2F49 CDC12F            CALL    WRSEC           ;Write Sector
              ;
              ;        Make a duplicate of sector Ø in sector 1
              ;
2F4C 1C                INC     E               ;Sector 1
2F4D CDC12F            CALL    WRSEC           ;Write sector
              ;
              ;        Write C/R in Auto Buffer in Sector 2
              ;
2F5Ø 1EØ2              LD      E,2             ;Sector 2
2F52 2E2Ø              LD      L,2ØH           ;Byte X'2Ø'
2F54 36ØD              LD      (HL),CR         ;No auto
2F56 CDC12F            CALL    WRSEC           ;Write sector
2F59 C9                RET                     ;RETurn for now
              ;
              ;        WRGAT - Write Granule Allocation Table
              ;
              ;
2F5A 21ØØ38   WRGAT    LD      HL,IOBUFF       ;HL => I/O buffer
2F5D 36F9     GATØ     LD      (HL),ØF9H       ;DD - X'F9', SD - X'FD'
2F5F 23                INC     HL              ;Bump
              ;
              ;        Lock out next X'CA' bytes in GAT
              ;
2F6Ø Ø6CA              LD      B,ØCAH          ;Lock out the bytes
2F62 36FF     LOCKOUT  LD      (HL),ØFFH       ;GAT + X'Ø1' through
2F64 23                INC     HL              ;GAT + X'CA'
2F65 1ØFB              DJNZ    LOCKOUT
              ;
              ;        GAT + X'CB'
              ;
2F67 3662              LD      (HL),62H        ;GAT + X'CB'= Version 6.2
              ;
              ;        GAT + X'CC'
              ;
2F69 3EØØ     CYLS     LD      A,$-$           ;P/u cylinder count
2F6B F5                PUSH    AF              ;Save Cylinder count
2F6C D623              SUB     35              ;Tracks in excess of 35
2F6E 23                INC     HL              ;HL => next GAT byte
2F6F 77                LD      (HL),A          ;GAT + X'CC'= tracks - 35
              ;
              ;        GAT + X'CD'
              ;
2F7Ø 23                INC     HL              ;GAT + X'CD' =
```

MEMDISKA - Installation

```
2F71 3642       1113Ø GATCD   LD      (HL),42H        ;DDEN, 1 side, 3 gran/cyl
                1114Ø ;
                1115Ø ;       GAT + X'CE' & X'CF'
                1116Ø ;
2F73 23         1117Ø         INC     HL              ;GAT + X'CE' & X'CF' =
2F74 36EØ       1118Ø         LD      (HL),ØEØH       ;16-bit Hash code of
2F76 23         1119Ø         INC     HL              ;"PASSWORD"
2F77 3642       112ØØ         LD      (HL),42H        ;Hash = X'42EØ'
                1121Ø ;
                1122Ø ;       GAT + X'DØ' - X'D7'
                1123Ø ;
2F79 23         1124Ø         INC     HL              ;HL => next GAT byte
2F7A 11B834     1125Ø         LD      DE,MEMDISK      ;"MEMDISK " is Pack name
2F7D ØEØ8       1126Ø         LD      C,8             ;Eight bytes
2F7F EB         1127Ø         EX      DE,HL           ;Swap 'em for LDIR
2F8Ø EDBØ       1128Ø         LDIR                    ;Stuff in ID
2F82 EB         1129Ø         EX      DE,HL           ;HL => GAT + X'D8'
                113ØØ ;
                1131Ø ;       GAT + X'D8' - X'DF'
                1132Ø ;
2F83            1133Ø         @@DATE                  ;Stuff date in GAT
2F83 3E12       ØØØ41         LD      A,18
2F85 EF         ØØØ42         RST     4Ø
                1134Ø ;
                1135Ø ;       Stuff GAT tracks in use with either X'F8' or X'FC'
                1136Ø ;
2F86 3EF8       1137Ø GPC     LD      A,ØF8H          ;3 gran/cyl
2F88 21Ø238     1138Ø         LD      HL,IOBUFF+2     ;HL => GAT + X'Ø2'
2F8B C1         1139Ø         POP     BC              ;B = # cylinders
2F8C Ø5         114ØØ         DEC     B               ;Subtract 2 to account
2F8D Ø5         1141Ø         DEC     B               ;For BOOT and DIR
                1142Ø ;
                1143Ø ;       Stuff open cylinder bytes into GAT
                1144Ø ;
2F8E 77         1145Ø FREETRK LD      (HL),A          ;Free track
2F8F 23         1146Ø         INC     HL              ;Next GAT byte
2F9Ø 1ØFC       1147Ø         DJNZ    FREETRK         ;Do it B times
                1148Ø ;
                1149Ø ;       Put 2 free Cyl bytes in lockout - BOOT & DIR
                115ØØ ;
2F92 2E6Ø       1151Ø         LD      L,6ØH           ;HL => Lockout
2F94 77         1152Ø         LD      (HL),A
2F95 2C         1153Ø         INC     L
2F96 77         1154Ø         LD      (HL),A
                1155Ø ;
                1156Ø ;       GAT + X'62' - GAT + X'BF'
                1157Ø ;
2F97 2EØ2       1158Ø         LD      L,2             ;HL => GAT + X'Ø2'
2F99 54         1159Ø         LD      D,H             ;Xfer to DE
2F9A 5D         116ØØ         LD      E,L
2F9B ØE6Ø       1161Ø         LD      C,6ØH           ;Of X'6Ø' for the
2F9D Ø9         1162Ø         ADD     HL,BC           ;  duplicate of top
2F9E ØD         1163Ø         DEC     C               ;Only duplicate X'5E'
2F9F ØD         1164Ø         DEC     C               ;  bytes
2FAØ EB         1165Ø         EX      DE,HL           ;Prepare for LDIR
2FA1 EDBØ       1166Ø         LDIR                    ;HL => GAT, DE => Lockout
                1167Ø ;
2FA3 11F438     1168Ø         LD      DE,IOBUFF+255-11        ;6.2 Media Data Block
2FA6 21BA2F     1169Ø         LD      HL,LSIID        ;Point to header
```

MEMDISKA - Installation

```
2FA9 010400                LD      BC,04           ;Set length
2FAC EDB0                  LDIR                    ;Move it
2FAE 2ABA32                LD      HL,(SAVEDCT)    ;The data to move
2FB1 23                    INC     HL
2FB2 23                    INC     HL
2FB3 23                    INC     HL
2FB4 0E07                  LD      C,7             ;Bytes to move
2FB6 EDB0                  LDIR                    ;Move it in
2FB8 1804                  JR      WRGAT1          ;Skip around string
2FBA 03            LSIID   DB      03,'LSI'
     4C 53 49
                   ;
2FBE 110001        WRGAT1  LD      DE,100H         ;D = Cyl 1, E = Sector 0
                   ;
                   ;       WRSEC - Write A sector to MemDISK drive
                   ;
2FC1 210038        WRSEC   LD      HL,IOBUFF       ;I/O buffer
2FC4 0E00          DRIVE   LD      C,$-$           ;P/u drive #
2FC6                       @@WRSEC                 ;Write Sector
2FC6 3E35                  LD      A,53
2FC8 EF                    RST     40
2FC9 C9                    RET                     ;  and RETurn
                   ;
                   ;       RDSEC - Read A sector of MemDISK drive
                   ;
2FCA 210038        RDSEC   LD      HL,IOBUFF       ;HL => I/O Buffer
2FCD 3AC52F                LD      A,(DRIVE+1)     ;P/u drive #
2FD0 4F                    LD      C,A             ;Xfer to C
2FD1                       @@RDSEC                 ;Read sector
2FD1 3E31                  LD      A,49
2FD3 EF                    RST     40
2FD4 C9                    RET                     ;  and RETurn
                   ;
                   ;       WRHIT - Write HIT sector in directory
                   ;
2FD5 AF            WRHIT   XOR     A               ;Set A = 0
2FD6 77            ZEROHIT LD      (HL),A          ;Zero HIT position
2FD7 2C                    INC     L               ;Bump HIT pointer
2FD8 20FC                  JR      NZ,ZEROHIT      ;256 positions
2FDA 36A2                  LD      (HL),0A2H       ;Hash for BOOT/SYS
2FDC 2C                    INC     L               ;HL => HIT + X'01'
2FDD 36C4                  LD      (HL),0C4H       ;Hash for DIR/SYS
2FDF 1C                    INC     E               ;D = Cyl 1, Sector 1
2FE0 18DF                  JR      WRSEC           ;Write Sector & RETurn
                   ;
                   ;       WRENT - Write DIR/SYS & BOOT/SYS entries
                   ;
2FE2 11FF2F        WRENT   LD      DE,BOOT         ;BOOT/SYS byte field
2FE5 EB                    EX      DE,HL           ;Swap for LDIR
2FE6 012000                LD      BC,32           ;32 bytes in entry
2FE9 EDB0                  LDIR                    ;Block move
2FEB 110201                LD      DE,102H         ;D = Cyl 1, E = Sector 2
2FEE CDC12F                CALL    WRSEC           ;Write Sector
                   ;
2FF1 012000                LD      BC,32
2FF4 EB                    EX      DE,HL           ;Xfer buffer ptr to DE
2FF5 211F30                LD      HL,DIR          ;HL => DIR/SYS bytes
2FF8 EDB0                  LDIR                    ;Xfer to MemDISK
2FFA 110301                LD      DE,103H         ;D = Cyl 1, E = Sector 3
```

MEMDISKA - Installation

```
2FFD 18C2                  JR      WRSEC           ;Write sector & RETurn
                   ;
                   ;       BOOT/SYS directory entry data
                   ;
2FFF 5E            BOOT    DB      Ø1Ø1111ØB       ;No access,inv,sys,FPDE
3ØØØ ØØØØ                  DW      Ø               ;Date = ØØ/ØØ/ØØ
3ØØ2 ØØØØ                  DW      Ø               ;EOF offset = Ø, LRL=256
3ØØ4 42                    DB      'BOOT    '      ;Name field
     4F 4F 54 2Ø 2Ø 2Ø 2Ø
3ØØC 53                    DB      'SYS'           ;Extension
     59 53
3ØØF F637                  DW      Ø37F6H          ;Owner password hash
3Ø11 F59C                  DW      Ø9CF5H          ;User password hash
3Ø13 Ø6ØØ          BOOTERN DW      6               ;ERN = 6 or 5
3Ø15 ØØ                    DB      Ø               ;First extent = Cyl Ø
3Ø16 ØØ            BOOTGRN DB      Ø               ;St gran = Ø, 1 cont gran
3Ø17 FFFF                  DW      ØFFFFH          ;No more extents
3Ø19 FFFF                  DW      ØFFFFH
3Ø1B FFFF                  DW      ØFFFFH
3Ø1D FFFF                  DW      ØFFFFH
                   ;
                   ;       DIR/SYS directory entry data
                   ;
3Ø1F 5D            DIR     DB      Ø1Ø111Ø1B       ;Read only,inv,sys,FPDE
3Ø2Ø ØØØØ                  DW      Ø               ;Date= ØØ/ØØ/ØØ
3Ø22 ØØØØ                  DW      Ø               ;EOF offset=Ø, LRL=256
3Ø24 44                    DB      'DIR     '      ;Name field
     49 52 2Ø 2Ø 2Ø 2Ø 2Ø
3Ø2C 53                    DB      'SYS'           ;Extension
     59 53
3Ø2F F637                  DW      Ø37F6H          ;Owner password hash
3Ø31 9642                  DW      Ø4296H          ;User password hash
3Ø33 12ØØ          DIRERN  DW      18              ;ERN+1 = 1Ø or 18
3Ø35 Ø1                    DB      1               ;Starts on cylinder 1
3Ø36 Ø2            SDENI   DB      ØØØØØØ1ØB       ;St. gran=Ø, 3 cont grans
3Ø37 FFFF                  DW      ØFFFFH          ;No Second Extent
3Ø39 FFFF                  DW      ØFFFFH          ;No Third Extent
3Ø3B FFFF                  DW      ØFFFFH          ;No Fourth Extent
3Ø3D FF                    DB      ØFFH            ;No further records
3Ø3E FF                    DB      ØFFH
                   ;
                   ;       DOMEM - Issue Prompts & take inputs for type
                   ;
3Ø3F 21E732        DOMEM   LD      HL,HELLO$       ;Display message
3Ø42                       @@DSPLY
                           IFEQ    ØØH,1
                           LD      HL,
                           ENDIF
3Ø42 3EØA                  LD      A,1Ø
3Ø44 EF                    RST     4Ø
                   ;
                   ;       Check if entry from SYSTEM (DRIVER= command
                   ;
3Ø45                       @@FLAGS
3Ø45 3E65                  LD      A,1Ø1
3Ø47 EF                    RST     4Ø
3Ø48 FDCBØ25E              BIT     3,(IY+'C'-'A')  ;System request?
3Ø4C CAC632                JP      Z,VIASET        ;Quit if not
                   ;
```

MEMDISKA - Installation

```
                ;       Input MemDISK type - A,B,C,D or E to disable
                ;
304F 216D33     GETYPE  LD      HL,BANKS          ;Display prompt
3Ø52                    @@DSPLY
                        IFEQ    ØØH,1
                        LD      HL,
                        ENDIF
3Ø52 3EØA               LD      A,1Ø
3Ø54 EF                 RST     4Ø
3Ø55 Ø6Ø1               LD      B,1               ;# of chars to input
3Ø57 CDD62C             CALL    INPUT             ;Input byte
3Ø5A 28F3               JR      Z,GETYPE          ;<ENTER> ? - re-input
                ;
                ;       Convert input A-E to Ø-4
                ;
3Ø5C 7E                 LD      A,(HL)            ;P/u first character
3Ø5D CBAF               RES     5,A               ;Convert to U/C
3Ø5F D641               SUB     'A'               ;<A> - Bank Ø ?
3Ø61 32292C             LD      (SETBANK+1),A     ;Save type of MemDISK
3Ø64 4F                 LD      C,A               ;Xfer to C for @BANK
                ;
                ;       If input is illegal then re-input
                ;
3Ø65 38E8               JR      C,GETYPE          ;Less - re-input
3Ø67 FEØ4               CP      4                 ;<E> - Disable MemDISK
3Ø69 CA8A31             JP      Z,DISMEM          ;Yes - take it out
3Ø6C 3ØE1               JR      NC,GETYPE         ;>4 - Re-input
                ;
                ;       Check if MemDISK is already active
                ;
3Ø6E FDCBØ366           BIT     4,(IY+DFLAG$)     ;MemDISK already active ?
3Ø72 C2CA32             JP      NZ,MEMIN          ;Yes - abort
                ;
                ;       If Type A,B,C - Check Bk, D - Check bks 1&2
                ;
3Ø75 C5                 PUSH    BC                ;Save Bank #
3Ø76 FEØ3               CP      3                 ;Type "D" ?
3Ø78 2ØØ6               JR      NZ,A_B_C          ;No - "A", "B", or "C"
                ;
                ;       Type "D" - See if both banks 1 & 2 are avail
                ;
3Ø7A ØEØ1       TYPED   LD      C,1               ;Bank #1 active ?
3Ø7C CDCB2C             CALL    CKBANK
3Ø7F ØC                 INC     C                 ;Bank #2 active ?
3Ø8Ø CDCB2C     A_B_C   CALL    CKBANK
3Ø83 C1                 POP     BC                ;C = Bank # (Ø,1,2,3)
                ;
                ;       Stuff Default Bank # and offset into driver
                ;
3Ø84 79                 LD      A,C               ;P/u bank #
3Ø85 3D                 DEC     A                 ;If bank Ø requested,
3Ø86 FA983Ø             JP      M,WASØ            ;  then keep as -1
3Ø89 3C                 INC     A                 ;  for driver bank test
3Ø8A 32C82D             LD      (BANKIM),A        ;Save bank # in driver
3Ø8D FEØ2               CP      2                 ;Instruction if
3Ø8F 2ØØ5               JR      NZ,NOT2           ;Just bank #2 active
3Ø91 21662E             LD      HL,OFFSET+1       ;Stuff X'8Ø' in ADD
3Ø94 368Ø               LD      (HL),8ØH
3Ø96 3EØ1       NOT2    LD      A,1               ;Always init to bank 1
```

MEMDISKA - Installation

```
                                                  ;  if type B, C or D
3Ø98 326B2E      WASØ     LD      (DEFBANK+1),A   ;Stuff in driver
                 ;
                 ;        Input Density (Single or Double)
                 ;
3Ø9B 215234      INPDENS  LD      HL,DENSITY      ;"Density"
3Ø9E                      @@DSPLY
                          IFEQ    ØØH,1
                          LD      HL,
                          ENDIF
3Ø9E 3EØA                 LD      A,1Ø
3ØAØ EF                   RST     4Ø
3ØA1 Ø6Ø1                 LD      B,1             ;Input an "S" or "D"
3ØA3 CDD62C               CALL    INPUT
3ØA6 2856                 JR      Z,DEFAULT       ;<ENTER> - use default
                 ;
                 ;        <D>ouble Density input ?
                 ;
3ØA8 7E                   LD      A,(HL)          ;P/u first char
3ØA9 CBAF                 RES     5,A             ;Convert to U/C
3ØAB FE44                 CP      'D'             ;<D>ouble Density ?
3ØAD 284F                 JR      Z,DEFAULT       ;Yes - use 6 sectors/gran
                 ;
                 ;        <S>ingle Density input ?
                 ;
3ØAF FE53                 CP      'S'             ;<S>ingle Density ?
3ØB1 2ØE8                 JR      NZ,INPDENS      ;No - input density again
                 ;
                 ;        Single Density - Change driver math
                 ;
3ØB3 3E82                 LD      A,82H           ;ADD A,D instruction
3ØB5 32622E               LD      (SDENB),A
3ØB8 3E87                 LD      A,87H           ;ADD A,A instruction
3ØBA 32632E               LD      (SDENC),A
3ØBD 3EØ9                 LD      A,9
3ØBF 32B42D               LD      (SDENF+3),A     ;DCT + 7
3ØC2 326A31               LD      (SPC+1),A       ;Save in CALCSIZ routine
3ØC5 3C                   INC     A               ;SDEN BOOT ERN = 1Ø
3ØC6 32333Ø               LD      (DIRERN),A      ;SDEN DIR/SYS ERN = 1Ø
3ØC9 3E24                 LD      A,24H
3ØCB 32B82D               LD      (SDENG+3),A     ;DCT + 8
3ØCE 3E32                 LD      A,'2'           ;Change size to 2.5ØK
3ØDØ 322Ø34               LD      (FRTRK1),A      ;Space per cylinder
3ØD3 3EFD                 LD      A,ØFDH          ;1 Gran Free
3ØD5 325E2F               LD      (GATØ+1),A      ;Stuff in WRGAT routine
3ØD8 3D                   DEC     A               ;2 Grans/Cyl - X'FC'
3ØD9 32872F               LD      (GPC+1),A
3ØDC AF                   XOR     A               ;NOP instruction
3ØDD 325E2E               LD      (SDENA),A
3ØEØ 32A52D               LD      (SDEND+3),A     ;DCT + 3
3ØE3 3C                   INC     A               ;Set A = 1
3ØE4 32722F               LD      (GATCD+1),A     ;Stuff in WRGAT routine
3ØE7 32363Ø               LD      (SDENI),A       ;2 contiguous granules
3ØEA 3EØ5                 LD      A,5             ;Set Boot ERN = 5
3ØEC 32133Ø               LD      (BOOTERN),A
3ØEF 3E1Ø                 LD      A,1ØH           ;Alien Disk Controller
3ØF1 32A92D               LD      (SDENE+3),A
3ØF4 213B2F               LD      HL,BTSECS+1     ;HL => # BOOT sectors
3ØF7 35                   DEC     (HL)            ;Use 5 instead of 6
```

MEMDISKA - Installation

```
3ØF8 21ØØØA                  LD      HL,SDBPC          ;Change GETCYL routine
3ØFB 22E52C                  LD      (BPC+1),HL
                     ;
                     ;       Calculate # of possible cylinders
                     ;
3ØFE 3A292C          DEFAULT LD      A,(SETBANK+1)     ;P/u type of memdisk
31Ø1 4F                      LD      C,A               ;Save in C
31Ø2 B7                      OR      A                 ;Bank Ø ?
31Ø3 28ØA                    JR      Z,PIKUPHI         ;Yes - use HIGH$
                     ;
                     ;       Bank #1, #2, or #1 & #2
                     ;
31Ø5 21FF7F                  LD      HL,7FFFH          ;HL = # bytes in 1 bank
31Ø8 FEØ3                    CP      3                 ;Bank 1 & 2 ?
31ØA 2Ø1F                    JR      NZ,CALCYL         ;No - use X'7FFF'
31ØC 65                      LD      H,L               ;Set HL = X'FFFF'
31ØD 181C                    JR      CALCYL
                     ;
                     ;       Bank Zero request - calculate free mem avail
                     ;
31ØF AF              PIKUPHI XOR     A                 ;Set A = Ø
311Ø ED62                    SBC     HL,HL             ;HL = Ø
3112 47                      LD      B,A               ;B = Ø
3113                         @@HIGH$                   ;P/u HIGH$
3113 3E64                    LD      A,1ØØ
3115 EF                      RST     4Ø
3116 22C634                  LD      (MDDATA+2),HL     ;Save HIGH$
3119 22C62D                  LD      (OLD_HI),HL       ;Save HIGH$ in driver
311C 23                      INC     HL                ;Set HL = last page
311D 25                      DEC     H
311E 6F                      LD      L,A
311F 226F31                  LD      (SAVPAGE+1),HL    ;Save page boundary
3122 11ØØ8Ø                  LD      DE,LOWEST         ;DE = lowest
3125 AF                      XOR     A
3126 ED52                    SBC     HL,DE             ;HL = amount free
3128 DAD232                  JP      C,NOMEM           ;Carry - not enough mem
                     ;
                     ;       Calculate # of cylinders available
                     ;
312B CDE22C          CALCYL  CALL    GETCYL            ;Get # of poss cyls
312E C2D232                  JP      NZ,NOMEM          ;NZ - Not enough mem
                     ;
                     ;       Convert A to ASCII & stuff into string
                     ;
3131 3C                      INC     A                 ;Bump one
3132 325F31                  LD      (MAXCYL+1),A      ;Save max # of cyls
3135 3D                      DEC     A
3136 326A2F                  LD      (CYLS+1),A        ;Stuff in WRGAT routine
3139 F5                      PUSH    AF                ;Save Max # of cyls
313A CD632C                  CALL    DECASC            ;Convert to ASCII in HL
313D F1                      POP     AF                ;A = # cyls
313E EB                      EX      DE,HL             ;DE = #
313F 214C34                  LD      HL,FRTRK2         ;HL => Destination
3142 72                      LD      (HL),D            ;Msb
3143 23                      INC     HL
3144 73                      LD      (HL),E            ;Lsb
                     ;
                     ;       A = # of Cyls poss, put in string if bank Ø
                     ;
```

MEMDISKA - Installation

```
3145 0C                   INC     C                ;Bank Zero request ?
3146 0D                   DEC     C
3147 C0                   RET     NZ               ;No - done prompting
                 ;
                 ;        Display Cylinders string & input # of cyls
                 ;
3148 210534      REDO     LD      HL,FRTRACK       ;How many cylinders
314B                      @@DSPLY
                          IFEQ    00H,1
                          LD      HL,
                          ENDIF
314B 3E0A                 LD      A,10
314D EF                   RST     40
314E 0602                 LD      B,2              ;Input # of cyls
3150 CDD62C               CALL    INPUT
3153 28F3                 JR      Z,REDO           ;Reinput it
                 ;
                 ;        Check if input legal
                 ;
3155 CD6E2C               CALL    DECHEX           ;Convert # to Hex
3158 20EE                 JR      NZ,REDO          ;Illegal - Re-input
315A FE03                 CP      MINCYL           ;Less than minimum?
315C 38EA                 JR      C,REDO
315E FE00        MAXCYL   CP      $-$              ;P/u max # of cyls
3160 30E6                 JR      NC,REDO          ;Too many - reinput
3162 326A2F               LD      (CYLS+1),A       ;New # of cylinders
                 ;
                 ;        CALCSIZ - Calculate Size of Cyl request
                 ;
3165 CDAE2C      CALCSIZ  CALL    SAVEREG          ;Save Registers
3168 4F                   LD      C,A              ;Xfer # cyls to C
3169 0611        SPC      LD      B,17             ;P/u Sectors/Cyl
                 ;
                 ;        Multiply Sectors per Cylinder x # Cylinders
                 ;
316B 81          MLOOP    ADD     A,C              ;Multiply B x C
316C 10FD                 DJNZ    MLOOP
                 ;
                 ;        Set HL = New HIGH$
                 ;
316E 210000      SAVPAGE  LD      HL,$-$           ;P/u page boundary
3171 ED44                 NEG                      ;Set H = H - A
3173 84                   ADD     A,H
3174 67                   LD      H,A              ;HL = New HIGH$, B = 0
3175 32662E               LD      (OFFSET+1),A     ;Stuff into driver
                 ;
                 ;        Stuff a Memory Header on front of MemDISK
                 ;
3178 2B                   DEC     HL               ;Pt 1 byte before
3179 EB                   EX      DE,HL            ;  Memdisk himem area
317A 21D434               LD      HL,MDDATA+16     ;Pt to header block
317D 011100               LD      BC,17
3180 EDB8                 LDDR                     ;  and move it to himem
3182 EB                   EX      DE,HL
3183 22CB2D               LD      (MEMHIGH),HL
3186                      @@HIGH$                  ;Install new HIGH$
3186 3E64                 LD      A,100
3188 EF                   RST     40
3189 C9                   RET                      ;Restore Regs & RETurn
```

MEMDISKA - Installation

```
                 ;
                 ;       DISMEM - Disable MemDISK if in memory
                 ;
318A FDCB0366    DISMEM  BIT     4,(IY+DFLAG$)    ;MemDISK active ?
318E CAD632              JP      Z,NOTPRS         ;No - display error mess
                 ;
                 ;       Pick up Driver address of drive
                 ;
3191 2ABA32              LD      HL,(SAVEDCT)     ;P/u DCT address
3194 E5                  PUSH    HL               ;Save DCT ptr
3195 23                  INC     HL               ;P/u driver address
3196 5E                  LD      E,(HL)           ;Lsb
3197 23                  INC     HL
3198 56                  LD      D,(HL)           ;Msb
3199 D5                  PUSH    DE               ;Save Driver Address
                 ;
                 ;       Calculate end of driver & Posn to ID
                 ;
319A EB                  EX      DE,HL            ;Pt HL to driver
319B E5                  PUSH    HL               ;Save driver start
319C 01DC00              LD      BC,LENGTH        ;Add length of driver
319F 09                  ADD     HL,BC            ;  to start of driver.
31A0 22F831              LD      (DREND+1),HL     ;Save next available
31A3 E1                  POP     HL               ;HL => driver add start
31A4 23                  INC     HL               ;Pos'n to length byte
31A5 23                  INC     HL
31A6 23                  INC     HL
31A7 23                  INC     HL
                 ;
                 ;       P/u length byte & pt to driver name
                 ;
31A8 46                  LD      B,(HL)           ;P/u length byte
31A9 23                  INC     HL               ;HL => Driver Name
31AA 11C034              LD      DE,MD$           ;DE => MEMDISK
                 ;
                 ;       Is this REALLY a certified MemDISK ??
                 ;
31AD 1A          MEMLP   LD      A,(DE)           ;P/u MemDISK byte
31AE BE                  CP      (HL)             ;Match ?
31AF 23                  INC     HL               ;Bump driver ptr
31B0 13                  INC     DE               ;Bump string ptr
31B1 C2CE32              JP      NZ,NOTMEM        ;No - isn't a MemDISK
31B4 10F7                DJNZ    MEMLP            ;Yes - check all posns
                 ;
                 ;       Pick up Old HIGH$ address & stuff for later
                 ;
31B6 5E                  LD      E,(HL)           ;P/u old HIGH$
31B7 23                  INC     HL
31B8 56                  LD      D,(HL)
31B9 ED53EC31            LD      (SAVEOLD+1),DE   ;Stuff into LD HL inst
                 ;
                 ;       P/u BANK information
                 ;
31BD FDCB03A6            RES     4,(IY+DFLAG$)    ;Reset MemDISK bit
31C1 23                  INC     HL               ;HL => Bank image
31C2 7E                  LD      A,(HL)           ;P/u bank image
31C3 4F                  LD      C,A              ;Xfer to C
31C4 FE03                CP      3                ;Both banks 1 & 2 ?
31C6 3805                JR      C,FRBANK         ;No - free up bank
```

MEMDISKA - Installation

```
31C8 ØD                     DEC     C               ;Set C = 2
31C9 CD3C2C                 CALL    FREBANK         ;Free bank #2
31CC ØD                     DEC     C               ;Set C = 1
31CD CD3C2C       FRBANK    CALL    FREBANK         ;Free Bank in C
                  ;
                  ;         Is this a Bank Zero MemDISK ?
                  ;
31DØ FD215E37               LD      IY,TYPEDIS      ;IY => Disable Type
31D4 ØC                     INC     C               ;Is C = Ø ?
31D5 ØD                     DEC     C
31D6 2Ø1C                   JR      NZ,GTDRV2       ;No - check out driver
                  ;
                  ;         Bank Ø - p/u last HIGH$ from Driver storage
                  ;
31D8 FD35ØØ                 DEC     (IY)            ;Change type
31DB 23                     INC     HL              ;Pos to HI$ val after
31DC 23                     INC     HL              ;  MemDISK installation.
31DD 23                     INC     HL
31DE 5E                     LD      E,(HL)          ;P/u address
31DF 23                     INC     HL
31EØ 56                     LD      D,(HL)
                  ;
                  ;         Pick up Current HIGH$ & compare with other
                  ;
31E1 6Ø                     LD      H,B             ;Set HL = Ø
31E2 68                     LD      L,B
31E3                        @@HIGH$                 ;(B=Ø), p/u HIGH$
31E3 3E64                   LD      A,1ØØ
31E5 EF                     RST     4Ø
31E6 B7                     OR      A               ;Same ?
31E7 ED52                   SBC     HL,DE
31E9 2ØØ9                   JR      NZ,GTDRV2       ;NZ - Can't do it
                  ;
                  ;         Reset HIGH$ = original HIGH$
                  ;
31EB 21ØØØØ       SAVEOLD   LD      HL,$-$          ;P/u old HIGH$
31EE                        @@HIGH$                 ;Re-allocate space
31EE 3E64                   LD      A,1ØØ
31FØ EF                     RST     4Ø
31F1 FD34ØØ                 INC     (IY)            ;Change Type
                  ;
                  ;         Can the Driver area be re-allocated ?
                  ;
31F4 CD352D       GTDRV2    CALL    GTDRV           ;Get driver area
31F7 21ØØØØ       DREND     LD      HL,$-$          ;P/u driver address
31FA B7                     OR      A
31FB ED52                   SBC     HL,DE           ;Same ?
31FD E1                     POP     HL              ;HL => Driver Address
31FE 2Ø16                   JR      NZ,NORECLM      ;No - can't Reclaim
                  ;
                  ;         Stuff original Address into low driver ptr
                  ;
32ØØ DD74ØØ                 LD      (IX),H          ;Msb
32Ø3 DD75FF                 LD      (IX-1),L        ;Lsb
32Ø6 FD34ØØ                 INC     (IY)            ;Change type
32Ø9 FD34ØØ                 INC     (IY)
                  ;
                  ;         Clear out Driver
                  ;
```

MEMDISKA - Installation

```
320C Ø1DBØØ          LD      BC,LENGTH-1     ;BC = # of bytes clr
320F 36ØØ            LD      (HL),Ø          ;Null byte
3211 54              LD      D,H             ;Set DE = HL+1
3212 5D              LD      E,L
3213 13              INC     DE
3214 EDBØ            LDIR                    ;Clear area
                 ;
                 ;       Disable DCT slot
                 ;
3216 E1          NORECLM POP     HL              ;HL => DCT + Ø
3217 36C9            LD      (HL),ØC9H       ;Disable it
                 ;
                 ;       Calculate Start of Disable string
                 ;
3219 FDE5            PUSH    IY              ;Xfer to HL
321B E1              POP     HL
321C 4E              LD      C,(HL)          ;P/u type
321D CB21            SLA     C               ;Multiply by 2
321F Ø6ØØ            LD      B,Ø             ;BC = offset in table
3221 23              INC     HL              ;HL => Address Table
3222 Ø9              ADD     HL,BC           ;HL => Add of mess string
3223 5E              LD      E,(HL)          ;P/u Address
3224 23              INC     HL
3225 56              LD      D,(HL)
3226 EB              EX      DE,HL           ;HL => Disable message
3227                 @@LOGOT                 ;Log message
                     IFEQ    ØØH,1
                     LD      HL,
                     ENDIF
3227 3EØC            LD      A,12
3229 EF              RST     4Ø
322A C3212C          JP      EXIT            ;Go to exit routine
                 ;
                 ;       FORMAT - Format Memory
                 ;
322D 21BØ36      FORMAT  LD      HL,VERIFY       ;"Verifying RAM ..."
323Ø                 @@DSPLY                 ;Display it
                     IFEQ    ØØH,1
                     LD      HL,
                     ENDIF
323Ø 3EØA            LD      A,1Ø
3232 EF              RST     4Ø
3233 16ØØ            LD      D,ØØ            ;Track counter
                 ;
                 ;       Display Current Cylinder Formatting
                 ;
3235 7A          WIPELP  LD      A,D             ;Get track counter
3236 CD442C          CALL    DECASC2         ;Display Dec ASCII equiv.
                 ;
                 ;       Run 4 different bit tests on each cylinder
                 ;
3239 3EFF            LD      A,11111111B     ;All bits on
323B CD5B32          CALL    VERCYL          ;Verify track w/ bits on
323E 3E55            LD      A,Ø1Ø1Ø1Ø1B     ;Next pattern
324Ø CD5B32          CALL    VERCYL
3243 3EAA            LD      A,1Ø1Ø1Ø1ØB     ;Last pattern
3245 CD5B32          CALL    VERCYL
3248 3EØØ            LD      A,ØØØØØØØØB     ;All bits off
324A CD5B32          CALL    VERCYL          ;Verify track w/ bits off
```

MEMDISKA - Installation

```
                ;
                ;       Finished Formatting yet ?
                ;
324D 14                 INC     D                  ;Bump cylinder #
324E 7A                 LD      A,D
324F DDBE06             CP      (IX+6)             ;Finished ?
3252 20E1               JR      NZ,WIPELP          ;No - stop when max cyl
                ;
                ;       Finished Formatting - Display message
                ;
3254 21CA36             LD      HL,FORMCOM         ;"Formatting Complete"
3257                    @@DSPLY                    ;Print it
                        IFEQ    00H,1
                        LD      HL,
                        ENDIF
3257 3E0A               LD      A,10
3259 EF                 RST     40
325A C9                 RET                        ;Done formatting
                ;
                ;       VERCYL - Verify a cylinder of RAM
                ;
325B 210038     VERCYL  LD      HL,IOBUFF          ;HL => I/O buffer
325E 1E00               LD      E,0                ;Init to sector 0
                ;
                ;       Fill buffer with specified byte
                ;
3260 77         STUFLP  LD      (HL),A             ;Stuff into buffer
3261 2C                 INC     L                  ;Bump
3262 20FC               JR      NZ,STUFLP          ;256 bytes to fill
                ;
                ;       Write the sector & read it back
                ;
3264 F5         CYLP    PUSH    AF                 ;Save fill byte
3265 CDC12F             CALL    WRSEC              ;Write Sector
3268 CDCA2F             CALL    RDSEC              ;Read into other buff
326B F1                 POP     AF                 ;A = Fill byte
                ;
                ;       Check if sector read back has correct byte
                ;
326C BE         CKLP    CP      (HL)               ;Match ?
326D C28D2C             JP      NZ,ERROR           ;No  - error
3270 2C                 INC     L                  ;Done with sector ?
3271 20F9               JR      NZ,CKLP            ;256 bytes to check
                ;
                ;       Advance to next sector
                ;
3273 7B                 LD      A,E                ;P/u sector #
3274 DDBE07             CP      (IX+7)             ;Finished ?
3277 7E                 LD      A,(HL)             ;P/u cylinder byte
3278 13                 INC     DE                 ;Bump E
3279 20E9               JR      NZ,CYLP            ;DCT+8 sectors to check
327B C9                 RET                        ;Done - RETurn
                ;
                ;       FORMTIT - Check if MemDISK has data on it
                ;
327C 110001     FORMTIT LD      DE,100H            ;D = Cyl 1, Sec 0 (GAT)
327F CDCA2F             CALL    RDSEC              ;Read BOOT sector
                ;
                ;       Check GAT ID
```

MEMDISKA - Installation

```
             ;
3282 2ED0              LD      L,0D0H          ;MemDISK pack name
3284 11B834            LD      DE,MEMDISK      ;What it should be
3287 0608              LD      B,8             ;# of characters
             ;
3289 1A        CKMLP   LD      A,(DE)          ;P/u should be char
328A BE                CP      (HL)            ;Match ?
328B 23                INC     HL              ;Bump
328C 13                INC     DE
328D 200C              JR      NZ,NOMTCH       ;No - must format
328F 10F8              DJNZ    CKMLP           ;Yes - loop for more
             ;
             ;         Already a MemDISK - Sure about formatting ?
             ;
3291 217434            LD      HL,DOFORM       ;Destination ...
3294 3E01              LD      A,1             ;Set MemDISK in flag
3296 32EA2E            LD      (MEMIN1+1),A
3299 1803              JR      DISMES          ;Display it
             ;
             ;         Not a MemDISK - Do normal Prompt
             ;
329B 219634    NOMTCH  LD      HL,STILLFM      ;Do you wish to format ?
329E           DISMES  @@DSPLY                 ;Display message
                       IFEQ    00H,1
                       LD      HL,
                       ENDIF
329E 3E0A              LD      A,10
32A0 EF                RST     40
             ;
             ;         Input Response
             ;
32A1 0601              LD      B,1             ;Input 1 character
32A3 E5                PUSH    HL              ;Save message start
32A4 CDD62C            CALL    INPUT
32A7 7E                LD      A,(HL)          ;P/u character
32A8 E1                POP     HL              ;Recover message start
32A9 05                DEC     B               ;Anything entered ?
32AA C0                RET     NZ              ;No - RETurn NZ
             ;
             ;         Set Z flag if "Y" & Reset Z if "N" entered
             ;
32AB CBAF              RES     5,A             ;Cvt to U/C
32AD FE4E              CP      'N'             ;<N>o ?
32AF 2805              JR      Z,RESZF         ;RETurn NZ
32B1 FE59              CP      'Y'             ;<Y>es ?
32B3 C8                RET     Z               ;RETurn Z set
32B4 18E8              JR      DISMES          ;No - reprompt
32B6 B7        RESZF   OR      A               ;Reset Z flag
32B7 C9                RET                     ;  and RETurn
             ;
             ;         Variables used
32B8 0000      SAVEDE  DW      0
32BA 0000      SAVEDCT DW      0
32BC 0000      DRADD   DW      0
             ;
             ;         Informative Error Display & Abort Routine
             ;
32BE 21D534    NODRV   LD      HL,NODRV$
32C1 DD                DB      0DDH
```

MEMDISKA - Installation

```
32C2 21F334    BADDRV  LD      HL,BADDRV$
32C5 DD                DB      ØDDH
32C6 211Ø37    VIASET  LD      HL,VIASET$        ;Not via SYSTEM
32C9 DD                DB      ØDDH
32CA 219936    MEMIN   LD      HL,MEMIN$         ;Already installed
32CD DD                DB      ØDDH
32CE 213235    NOTMEM  LD      HL,NOTMEM$        ;Not a MemDISK
32D1 DD                DB      ØDDH
32D2 214D35    NOMEM   LD      HL,NOMEM$         ;Insufficient Memory
32D5 DD                DB      ØDDH
32D6 216235    NOTPRS  LD      HL,NOTPRS$        ;Not Present
32D9 DD                DB      ØDDH
32DA 216736    BNKUSE  LD      HL,BNKUSE$        ;Bank in use
32DD DD                DB      ØDDH
32DE 217635    NOTACT  LD      HL,NOTACT$        ;Cant Install
               ;
               ;       Log Error Message & Abort
               ;
32E1                   @@LOGOT                   ;Log error message
                       IFEQ    ØØH,1
                       LD      HL,
                       ENDIF
32E1 3EØC              LD      A,12
32E3 EF                RST     4Ø
32E4 C31B2C            JP      ABORT             ;Go to exit routine
               ;
32E7 4D        HELLO$  DB      'MEMDISK'
     45 4D 44 49 53 4B
32EE           *GET    CLIENT:3
               ;CLIENTS/ASM - File to establish sign-on headers
               ;
32EE 2Ø                DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
3318 2Ø                DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
               ;
332D 41                DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
3355 2Ø                DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
               ;
336D ØA        BANKS   DB      LF,'<A>  Bank Ø (Primary Memory)',LF
     3C 41 3E 2Ø 2Ø 42 61 6E
     6B 2Ø 3Ø 2Ø 28 5Ø 72 69
     6D 61 72 79 2Ø 4D 65 6D
     6F 72 79 29 ØA
```

MEMDISKA - Installation

```
338B 3C        1792Ø          DB       '<B>  Bank 1',LF
     42 3E 2Ø 2Ø 42 61 6E 6B
     2Ø 31 ØA
3397 3C        1793Ø          DB       '<C>  Bank 2',LF
     43 3E 2Ø 2Ø 42 61 6E 6B
     2Ø 32 ØA
33A3 3C        1794Ø          DB       '<D>  Banks 1 and 2',LF
     44 3E 2Ø 2Ø 42 61 6E 6B
     73 2Ø 31 2Ø 61 6E 64 2Ø
     32 ØA
33B6 3C        1795Ø          DB       '<E>  Disable MemDISK',LF,LF
     45 3E 2Ø 2Ø 44 69 73 61
     62 6C 65 2Ø 4D 65 6D 44
     49 53 4B ØA ØA
33CC 57        1796Ø          DB       'Which type of allocation - '
     68 69 63 68 2Ø 74 79 7Ø
     65 2Ø 6F 66 2Ø 61 6C 6C
     6F 63 61 74 69 6F 6E 2Ø
     2D 2Ø
33E7 3C        1797Ø          DB       '<A>, <B>, <C>, <D>, or <E> ? ',ETX
     41 3E 2C 2Ø 3C 42 3E 2C
     2Ø 3C 43 3E 2C 2Ø 3C 44
     3E 2C 2Ø 6F 72 2Ø 3C 45
     3E 2Ø 3F 2Ø Ø3
               1798Ø ;
34Ø5 4E        1799Ø FRTRACK DB    'Note: Each Cylinder equals '
     6F 74 65 3A 2Ø 45 61 63
     68 2Ø 43 79 6C 69 6E 64
     65 72 2Ø 65 71 75 61 6C
     73 2Ø
342Ø 34        18ØØØ FRTRK1  DB    '4.5ØK of space.',LF
     2E 35 3Ø 4B 2Ø 6F 66 2Ø
     73 7Ø 61 63 65 2E ØA
343Ø 4E        18Ø1Ø          DB       'Number of free Cylinders: ',MINCYL+'Ø'&ØFFH,'-'
     75 6D 62 65 72 2Ø 6F 66
     2Ø 66 72 65 65 2Ø 43 79
     6C 69 6E 64 65 72 73 3A
     2Ø 33 2D
344C 3Ø        18Ø2Ø FRTRK2  DB    'ØØ ? ',ETX
     3Ø 2Ø 3F 2Ø Ø3
               18Ø3Ø ;
3452 53        18Ø4Ø DENSITY DB    'Single or Double Density <S,D> ? ',ETX
     69 6E 67 6C 65 2Ø 6F 72
     2Ø 44 6F 75 62 6C 65 2Ø
     44 65 6E 73 69 74 79 2Ø
     3C 53 2C 44 3E 2Ø 3F 2Ø
     Ø3
               18Ø5Ø ;
3474 44        18Ø6Ø DOFORM  DB    'Destination MemDISK contains Data',LF
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 4D 65 6D 44 49
     53 4B 2Ø 63 6F 6E 74 61
     69 6E 73 2Ø 44 61 74 61
     ØA
               18Ø7Ø ;
3496 44        18Ø8Ø STILLFM DB    'Do you wish to Format it <Y/N> ? ',ETX
     6F 2Ø 79 6F 75 2Ø 77 69
     73 68 2Ø 74 6F 2Ø 46 6F
     72 6D 61 74 2Ø 69 74 2Ø
```

MEMDISKA - Installation

```
     3C 59 2F 4E 3E 2Ø 3F 2Ø
     Ø3
               18Ø9Ø ;
34B8 4D        181ØØ MEMDISK DB        'MEMDISK '
     45 4D 44 49 53 4B 2Ø
34CØ 24        1811Ø MD$     DB        '$MD',ETX
     4D 44 Ø3
34C4 18        1812Ø MDDATA  DB        18H,17,Ø,Ø,8,'MemDISKD',Ø,Ø,Ø,Ø
     11 ØØ ØØ Ø8 4D 65 6D 44
     49 53 4B 44 ØØ ØØ ØØ ØØ
               1813Ø ;
34D5 4C        1814Ø NODRV$  DB        'Logical drive number required',CR
     6F 67 69 63 61 6C 2Ø 64
     72 69 76 65 2Ø 6E 75 6D
     62 65 72 2Ø 72 65 71 75
     69 72 65 64 ØD
34F3 43        1815Ø BADDRV$ DB        'Can''t specify SYSTEM drive slot',CR
     61 6E 27 74 2Ø 73 7Ø 65
     63 69 66 79 2Ø 53 59 53
     54 45 4D 2Ø 64 72 69 76
     65 2Ø 73 6C 6F 74 ØD
3513 4D        1816Ø INSTALD DB        'MemDISK Successfully Installed',CR
     65 6D 44 49 53 4B 2Ø 53
     75 63 63 65 73 73 66 75
     6C 6C 79 2Ø 49 6E 73 74
     61 6C 6C 65 64 ØD
               1817Ø ;
3532 54        1818Ø NOTMEM$ DB        'Target Drive not a MemDISK',CR
     61 72 67 65 74 2Ø 44 72
     69 76 65 2Ø 6E 6F 74 2Ø
     61 2Ø 4D 65 6D 44 49 53
     4B ØD
               1819Ø ;
354D 49        182ØØ NOMEM$  DB        'Insufficient Memory ',CR
     6E 73 75 66 66 69 63 69
     65 6E 74 2Ø 4D 65 6D 6F
     72 79 2Ø ØD
               1821Ø ;
3562 4D        1822Ø NOTPRS$ DB        'MemDISK not present',CR
     65 6D 44 49 53 4B 2Ø 6E
     6F 74 2Ø 7Ø 72 65 73 65
     6E 74 ØD
               1823Ø ;
3576 4D        1824Ø NOTACT$ DB        'MemDISK not present, installation '
     65 6D 44 49 53 4B 2Ø 6E
     6F 74 2Ø 7Ø 72 65 73 65
     6E 74 2C 2Ø 69 6E 73 74
     61 6C 6C 61 74 69 6F 6E
     2Ø
3598 61        1825Ø         DB        'aborted',CR
     62 6F 72 74 65 64 ØD
               1826Ø ;
35AØ 4D        1827Ø DISABE1 DB        'MemDISK disabled, memory now avail'
     65 6D 44 49 53 4B 2Ø 64
     69 73 61 62 6C 65 64 2C
     2Ø 6D 65 6D 6F 72 79 2Ø
     6E 6F 77 2Ø 61 76 61 69
     6C
35C2 61        1828Ø         DB        'able',CR
```

MEMDISKA - Installation

```
     62 6C 65 ØD
                 1829Ø ;
35C7 4D          183ØØ DISABE2 DB       'MemDISK disabled, Unable to reclaim '
     65 6D 44 49 53 4B 2Ø 64
     69 73 61 62 6C 65 64 2C
     2Ø 55 6E 61 62 6C 65 2Ø
     74 6F 2Ø 72 65 63 6C 61
     69 6D 2Ø
35EB 68          1831Ø         DB       'high memory',CR
     69 67 68 2Ø 6D 65 6D 6F
     72 79 ØD
                 1832Ø ;
35F7 4D          1833Ø DISABE3 DB       'MemDISK disabled, Unable to reclaim '
     65 6D 44 49 53 4B 2Ø 64
     69 73 61 62 6C 65 64 2C
     2Ø 55 6E 61 62 6C 65 2Ø
     74 6F 2Ø 72 65 63 6C 61
     69 6D 2Ø
361B 64          1834Ø         DB       'driver area',CR
     72 69 76 65 72 2Ø 61 72
     65 61 ØD
                 1835Ø ;
3627 4D          1836Ø DISABE4 DB       'MemDISK disabled, Unable to reclaim '
     65 6D 44 49 53 4B 2Ø 64
     69 73 61 62 6C 65 64 2C
     2Ø 55 6E 61 62 6C 65 2Ø
     74 6F 2Ø 72 65 63 6C 61
     69 6D 2Ø
364B 68          1837Ø         DB       'high memory and driver area',CR
     69 67 68 2Ø 6D 65 6D 6F
     72 79 2Ø 61 6E 64 2Ø 64
     72 69 76 65 72 2Ø 61 72
     65 61 ØD
                 1838Ø ;
3667 55          1839Ø BNKUSE$ DB       'Unable to install MemDISK, '
     6E 61 62 6C 65 2Ø 74 6F
     2Ø 69 6E 73 74 61 6C 6C
     2Ø 4D 65 6D 44 49 53 4B
     2C 2Ø
3682 72          184ØØ         DB       'requested bank in use.',CR
     65 71 75 65 73 74 65 64
     2Ø 62 61 6E 6B 2Ø 69 6E
     2Ø 75 73 65 2E ØD
                 1841Ø ;
3699 4D          1842Ø MEMIN$  DB       'MemDISK already Active',CR
     65 6D 44 49 53 4B 2Ø 61
     6C 72 65 61 64 79 2Ø 41
     63 74 69 76 65 ØD
                 1843Ø ;
36BØ 56          1844Ø VERIFY  DB       'Verifying RAM cylinder ØØ',ETX
     65 72 69 66 79 69 6E 67
     2Ø 52 41 4D 2Ø 63 79 6C
     69 6E 64 65 72 2Ø 3Ø 3Ø
     Ø3
                 1845Ø ;
36CA ØA          1846Ø FORMCOM DB       LF,'Verifying Complete, RAM good',LF
     56 65 72 69 66 79 69 6E
     67 2Ø 43 6F 6D 7Ø 6C 65
     74 65 2C 2Ø 52 41 4D 2Ø
```

MEMDISKA - Installation

```
     67 6F 6F 64 ØA
36E8 44          1847Ø          DB      'Directory has been placed on Cylinder 1',CR
     69 72 65 63 74 6F 72 79
     2Ø 68 61 73 2Ø 62 65 65
     6E 2Ø 7Ø 6C 61 63 65 64
     2Ø 6F 6E 2Ø 43 79 6C 69
     6E 64 65 72 2Ø 31 ØD
                 1848Ø ;
371Ø 4D          1849Ø VIASET$ DB      'Must install via SYSTEM (DRIVER=',CR
     75 73 74 2Ø 69 6E 73 74
     61 6C 6C 2Ø 76 69 61 2Ø
     53 59 53 54 45 4D 2Ø 28
     44 52 49 56 45 52 3D ØD
                 185ØØ ;
3731 ØA          1851Ø BADRAM  DB      LF,'Verify Error in Bank '
     56 65 72 69 66 79 2Ø 45
     72 72 6F 72 2Ø 69 6E 2Ø
     42 61 6E 6B 2Ø
3747 6E          1852Ø VBANK   DB      'n at location X',AP
     2Ø 61 74 2Ø 6C 6F 63 61
     74 69 6F 6E 2Ø 58 27
3757 6E          1853Ø VLOC    DB      'nnnn',AP,LF,CR
     6E 6E 6E 27 ØA ØD
                 1854Ø ;
375E Ø1          1855Ø TYPEDIS DB      1               ;Type of disable
375F 2736        1856Ø DISTAB  DW      DISABE4,DISABE3,DISABE2,DISABE1
     F735 C735 AØ35
3767 ØØ          1857Ø RE_USE  DB      Ø               ;Re-use trapped driver area.
                 1858Ø ;
                 1859Ø ;       Buffers Used
                 186ØØ ;
38ØØ             1861Ø         ORG     $<-8+1<+8
                 1862Ø ;
Ø1ØØ             1863Ø IOBUFF  DS      256
Ø1ØØ             1864Ø BUFFER  DS      256
ØØØA             1865Ø DUPDCT  DS      1Ø
                 1866Ø ;
                 ØØ57Ø ;
3AØA             ØØ58Ø         SUBTTL  <>
2CØØ             ØØ59Ø         END     START
```

```
$NOT          2F10 @@1          0000 @@2          0000
@@3           0000 @@4          0000 @MOD2        0000
@MOD4         FFFF ABB          0010 ABORT        2C1B
AP            0027 A_B_C        3080 B10          2E22
B13           2E29 B14          2E4B B14A         2E57
B9            2DF7 BADDRV       32C2 BADDRV$      34F3
BADRAM        3731 BANKIM       2DC8 BANKS        336D
BNKUSE        32DA BNKUSE$      3667 BOOT         2FFF
BOOTERN       3013 BOOTGRN      3016 BPC          2CE4
BREAK         0080 BS           0008 BTLP         2F3C
BTSECS        2F3A BUFF         2E8B BUFFER       3900
BUFFER$       2300 CALCDRV      2CF6 CALCSIZ      3165
CALCYL        312B CFLAG$       0002 CHKDIR       2E1A
CHKDIR2       2E19 CKBANK       2CCB CKLP         326C
CKMLP         3289 CR           000D CYLP         3264
CYLS          2F69 DDBPC        1200 DECASC       2C63
DECASC2       2C44 DECHEX       2C6E DEFAULT      30FE
DEFBANK       2E6A DENSITY      3452 DFLAG$       0003
DIR           301F DIRERN       3033 DISABE1      35A0
DISABE2       35C7 DISABE3      35F7 DISABE4      3627
DISMEM        318A DISMES       329E DISTAB       375F
DIVLP         2CE9 DIVLP1       2D13 DOFORM       3474
DOFORM1       2F13 DOMEM        303F DONE1        2C80
DONTRES       2EFA DOXFER       2E0C DOXFER1      2D26
DO_INST       2EDC DRADD        32BC DREND        31F7
DRIVE         2FC4 DRIVER       2DBE DRVLOW       2DC9
DSP           2C59 DUPDCT       3A00 ERROR        2C8D
ETX           0003 EX1          2E5B EXIT         2C21
FILBUF        2F34 FLAG         0040 FORMAT       322D
FORMCOM       36CA FORMTIT      327C FRBANK       31CD
FREBANK       2C3C FREETRK      2F8E FRTRACK      3405
FRTRK1        3420 FRTRK2       344C GAT0         2F5D
GATCD         2F71 GETADR       2E5D GETBUF       2E8A
GETCYL        2CE2 GETDIG       2C82 GETDUP       2D2C
GETOLD        2E83 GETYPE       304F GOTBANK      2E70
GPC           2F86 GTDRV        2D35 GTDRV2       31F4
HELLO$        32E7 HIDRVR       1300 ILLEGAL      2C8A
INIT          2DDD INPDENS      309B INPUT        2CD6
INSTALD       3513 INSTDRV      2D48 INSTMEM      2E9A
IOBUFF        3800 IOERR        2C5D KFLAG$       000A
KIDCB$        2EF7 LENGTH       00DC LF           000A
LOCKOUT       2F62 LOWEST       8000 LPADD        2C65
LSIID         2FBA MAXCYL       315E MD$          34C0
MDDATA        34C4 MEMDISK      34B8 MEMDRIV      2DF6
MEMHIGH       2DCB MEMIN        32CA MEMIN$       3699
MEMIN1        2EE9 MEMLP        31AD MINCYL       0003
MLOOP         316B MYSTACK      2DDD NODRV        32BE
NODRV$        34D5 NOMEM        32D2 NOMEM$       354D
NOMTCH        329B NORECLM      3216 NORMEX       2C16
NOT2          3096 NOTACT       32DE NOTACT$      3576
NOTDIR        2E20 NOTMEM       32CE NOTMEM$      3532
NOTPRS        32D6 NOTPRS$      3562 NOT__IN      2EB5
NUM           0080 OFFSET       2E65 OKTOGO       2ED6
OLDHIGH       2DC0 OLDRVR       2EEE OLD_HI       2DC6
PAR_ERR       002C PIKUPHI      310F RDSEC        2FCA
RECVDE        2E3F REDO         3148 REL1         2DFF
REL2          2E0F REL2A        2E06 REL3         2E40
REL4          2E46 REL5         2E79 REL6         2DE0
```

```
REL7             2DE5 REL8            2DEB REL8A           2E35
REL8B            2E7C REL9            2DE8 RELDUN          2D8A
RELTBL           2D7Ø RESTREG         2CC6 RESZF           32B6
RETADDR          2CC3 RE_USE          3767 RLOOP           2D57
SAVDCT           2D2Ø SAVEDCT         32BA SAVEDE          32B8
SAVEOLD          31EB SAVEREG         2CAE SAVESP          2DEE
SAVPAGE          316E SDBPC           ØAØØ SDENA           2E5E
SDENB            2E62 SDENC           2E63 SDEND           2DA2
SDENE            2DA6 SDENF           2DB1 SDENG           2DB5
SDENI            3Ø36 SETBANK         2C28 SETDCT          2D94
SFLAG$           ØØ12 SHOWINU         2F22 SPC             3169
START            2CØØ STARTA          2CØ9 STBANK          2C34
STFRET           2E72 STILLFM         3496 STR             ØØ2Ø
STUFLP           326Ø TAB             ØØØ9 TYPED           3Ø7A
TYPEDIS          375E VBANK           3747 VERCYL          325B
VERIFY           36BØ VFLAG$          ØØ15 VIASET          32C6
VIASET$          371Ø VLOC            3757 WASØ            3Ø98
WIPELP           3235 WP              ØØØF WRBOOT          2F3Ø
WRENT            2FE2 WRGAT           2F5A WRGAT1          2FBE
WRHIT            2FD5 WRITES          2E2D WRSEC           2FC1
ZEROHIT          2FD6 @@ABORT         7Ø59 @@ADTSK         7ØEC
@@BANK           76Ø4 @@BKSP          72E4 @@BREAK         761A
@@CHNIO          7Ø44 @@CKBRKC        7668 @@CKDRV         714Ø
@@CKEOF          72F9 @@CKTSK         7ØD7 @@CLOSE         72CF
@@CLS            7652 @@CMNDI         7Ø83 @@CMNDR         7Ø98
@@CTL            6EA8 @@DATE          7Ø1A @@DCSTAT        717F
@@DEBUG          7ØC2 @@DECHEX        7584 @@DIRRD         74F1
@@DIRWR          75Ø6 @@DIV16         756F @@DIV8          755A
@@DODIR          7155 @@DSP           6E6C @@DSPLY         6FØC
@@ERROR          7ØAD @@EXIT          7Ø6E @@FEXT          745E
@@FLAGS          75EE @@FNAME         7473 @@FSPEC         7449
@@GATRD          74DC @@GATWR         751B @@GET           6E8Ø
@@GTDCB          749D @@GTDCT         7488 @@GTMOD         74B2
@@HDFMT          7227 @@HEX16         75C3 @@HEX8          75AE
@@HEXDEC         7599 @@HIGH$         75D8 @@INIT          72A5
@@KBD            6EE4 @@KEY           6E58 @@KEYIN         6EF8
@@KLTSK          712B @@LOAD          741F @@LOC           73ØE
@@LOF            7323 @@LOGER         6F43 @@LOGOT         6F58
@@MSG            6F8F @@MUL16         7545 @@MUL8          753Ø
@@OPEN           72BA @@PARAM         7ØØ5 @@PAUSE         6FFØ
@@PEOF           7338 @@POSN          734D @@PRINT         6FA4
@@PRT            6EBC @@PUT           6E94 @@RAMDIR        716A
@@RDSEC          71FD @@RDSSC         74C7 @@READ          7362
@@REMOV          729Ø @@RENAM         727B @@REW           7377
@@RMTSK          71Ø1 @@RPTSK         7116 @@RREAD         738C
@@RSLCT          71E8 @@RSTOR         71A9 @@RUN           7434
@@RWRIT          73A1 @@SEEK          71D3 @@SEEKSC        73B6
@@SKIP           73CB @@SLCT          7194 @@STEPI         71BE
@@TIME           7Ø2F @@VDCTL         6FDB @@VER           73EØ
@@VRSEC          7212 @@WEOF          73F5 @@WHERE         6EDØ
@@WRITE          74ØA @@WRSEC         723C @@WRSSC         7251
@@WRTRK          7266
2CØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

PATCH/CMD - Disk file patch utility
Patch allows changing bytes in any type of disk file, be it a load module format file or standard data file. Patch code may be typed in on the command line or read from an ASCII disk file.

```
             ;PATCH/ASM
0000                 TITLE   <PATCH - LS-DOS 6.2>
             ;
0003         ETX     EQU     3
000A         LF      EQU     10
000D         CR      EQU     13
0040         FLAG    EQU     01000000B
0010         ABB     EQU     00010000B
             ;
0000         *GET    SVCMAC:3                ;SVC Macro equivalents
             ;SVCMAC/ASM - LS-DOS Version VI
             *LIST   OFF
             *LIST   ON
0000         *GET    COPYCOM:3               ;Copyright message
             ; COPYCOM - File for Copyright COMment block
             ;
0000                 COM     '<*(C) 1982,83,84 by LSI*>'
             ;
2600                 ORG     2600H
             ;
             BEGIN
2600                 @@CKBRKC                ;Check if Break hit
2600 3E6A            LD      A,106
2602 EF              RST     40
2603 2804            JR      Z,BEGINA        ;Continue if no break
2605 21FFFF          LD      HL,-1           ;  else abort
2608 C9              RET
             ;
             BEGINA
2609 ED73BF27        LD      (STACK),SP      ;Save original stack
260D E5              PUSH    HL              ;Save ptr to CMD buffer
260E                 @@FLAGS                 ;Set up IY
260E 3E65            LD      A,101
2610 EF              RST     40
2611 21C02D          LD      HL,HELLO$
2614 CD1F2D          CALL    $DSPLY          ;Display the signon msg
             ;
             ;       Get /CMD file off command line
             ;
2617 E1              POP     HL              ;P/u cmd line ptr
2618 117F2D          LD      DE,PGMDCB       ;Set up for OPEN
261B                 @@FSPEC                 ;Fetch program filespec
261B 3E4E            LD      A,78
261D EF              RST     40
261E C2512D          JP      NZ,PGMREQ       ;Quit if illegal name
2621 1A              LD      A,(DE)
2622 FE2A            CP      '*'             ;Test for device spec
2624 CA512D          JP      Z,PGMREQ        ;Abort if not a filespec
2627 E5              PUSH    HL              ;Save posn on command line
2628 21792D          LD      HL,CMDEXT
262B                 @@FEXT                  ;Default ext to /CMD
262B 3E4F            LD      A,79
262D EF              RST     40
262E D5              PUSH    DE              ;Save ptr to FCB
262F EB              EX      DE,HL           ;Pt HL at current name
2630 113330          LD      DE,FNM$         ;Store the name away
2633                 @@FSPEC                 ;  in case of a later error
2633 3E4E            LD      A,78
2635 EF              RST     40
2636 D1              POP     DE              ;Recover FCB
2637 210033          LD      HL,PGMBUF       ;Buffer for /CMD file I/O
```

```
263A Ø6ØØ                LD      B,Ø               ;Set lrl=256
263C CDEF2C              CALL    $OPEN             ;Open the file to fix
                 ;
                 ;       Get /FIX file (if any)
                 ;
263F E1                  POP     HL                ;Get command line posn
264Ø 11AØ2D              LD      DE,FIXDCB         ;FCB used for /FIX file
2643                     @@FSPEC                   ;See if a filespec is there
2643 3E4E                LD      A,78
2645 EF                  RST     4Ø
2646 C27426              JP      NZ,CKLIN          ;If error, ck for parms there
2649 E5                  PUSH    HL                ;Save command line posn
264A 217C2D              LD      HL,FIXEXT
264D                     @@FEXT                    ;Use default ext of /FIX
264D 3E4F                LD      A,79
264F EF                  RST     4Ø
265Ø 21AØ2D              LD      HL,FIXDCB         ;Pt HL to start of fix filespec
2653 11Ø93Ø              LD      DE,NAMFIX$        ;Buffer to hold filename only
2656 Ø6ØØ                LD      B,Ø               ;Init char count to Ø
                 ;
                 ;       Save patch file name for X header
                 ;
2658 7E          FXNAM   LD      A,(HL)            ;P/u a char of the filespec
2659 23                  INC     HL
265A FE2F                CP      '/'               ;Found the /FIX ext?
265C 2811                JR      Z,FXNAM2          ;Quit if so
265E FE3A                CP      ':'               ;Colon yet?
266Ø 38Ø8                JR      C,FXNAM1          ;If less, must be number
2662 FE41                CP      'A'               ;A-Z?
2664 38Ø9                JR      C,FXNAM2          ;If less, done
2666 FE5B                CP      'Z'+1             ;If not alpha, done
2668 3ØØ5                JR      NC,FXNAM2
266A 12          FXNAM1  LD      (DE),A            ;Store the name char
266B 13                  INC     DE                ;Inc storage ptr
266C Ø4                  INC     B                 ;Inc count of name chars
266D 18E9                JR      FXNAM             ;Loop for more
266F 78          FXNAM2  LD      A,B               ;Store the length of
267Ø 32Ø83Ø              LD      (NAMLEN$),A       ;  the /FIX patch file
2673 E1                  POP     HL                ;Recover command line posn
2674 7E          CKLIN   LD      A,(HL)            ;Test command line
2675 FEØD                CP      CR                ;  for end
2677 2845                JR      Z,RDFIX           ;Go if found
2679 23                  INC     HL
267A FE2Ø                CP      2ØH
267C 28F6                JR      Z,CKLIN           ;Ignore spaces
267E FE28                CP      '('               ;Beginning of parm?
268Ø C2492D              JP      NZ,PRMERR         ;Anything else is a parm error
                 ;
                 ;       Test for REMOVE or special Option parameters
                 ;       Ignore @@PARAM errors, as the parameters may actually
                 ;        be a command line patch.
                 ;
2683 11573Ø              LD      DE,PTBL$          ;Parameter table
2686 E5                  PUSH    HL                ;Save command line ptr
2687 2B                  DEC     HL                ;Back up to '('
2688                     @@PARAM
2688 3E11                LD      A,17
268A EF                  RST     4Ø
268B E1                  POP     HL                ;Restore cmd line ptr
268C Ø1ØØØØ              LD      BC,$-$            ;"Remove" parm response
268D             RPARM1  EQU     $-2
268F 79                          LD      A,C
2690 32472C                      LD      (RPARM),A       ;Set Remove parm
2693 01FFFF                      LD      BC,-1           ;0 parm - bypass need for
2694                     OPARM1  EQU     $-2             ;  Frr,nn line if OFF
2696 79                          LD      A,C
2697 32412C                      LD      (OPARM),A       ;Set find flag
269A CABE26                      JP      Z,RDFIX         ;If @PARAM was good, there is
                                                         ;  no cmd line patch code
                         ;
                         ;       Check for command line patch code (CLP)
                         ;
269D 010034                      LD      BC,FIXDATA      ;Space allocated for /FIX data
26A0 7E                  CKLIN1  LD      A,(HL)          ;Get char from cmd line
26A1 FE0D                        CP      CR
26A3 CAB526                      JP      Z,CKLIN3        ;Show end of CLP
26A6 FE29                        CP      ')'
26A8 280B                        JR      Z,CKLIN3        ;End of CLP if so
26AA 23                          INC     HL              ;Bump buffer ptr
26AB FE3A                        CP      ':'             ;Separator between patches?
26AD 2002                        JR      NZ,CKLIN2       ;If not, store char
26AF 3E0D                        LD      A,CR            ;  else show end of this CLP
26B1 02                  CKLIN2  LD      (BC),A          ;Put byte into fix data buff
26B2 03                          INC     BC              ;Bump buff ptr
26B3 18EB                        JR      CKLIN1          ;Loop til end of cmd line
                         ;
26B5 3E0D                CKLIN3  LD      A,CR            ;Put CR into
26B7 02                          LD      (BC),A          ;  CLP buffer
26B8 03                          INC     BC
26B9 3E03                        LD      A,ETX           ;End buffer with ETX
26BB 02                          LD      (BC),A
26BC 1839                        JR      DOFIX           ;Start patching...
                         ;
                         ;       P/u the fix info from the FIX file, rather than
                         ;        the command line.
                         ;
26BE 3A0830              RDFIX   LD      A,(NAMLEN$)     ;P/u len of /FIX filename
26C1 B7                          OR      A
26C2 CA512D                      JP      Z,PGMREQ        ;If none used, abort
26C5 FDCB12C6                    SET     0,(IY+'S'-'A')  ;Set open inhibit bit
26C9 11A02D                      LD      DE,FIXDCB       ;Set up & open /FIX file
26CC 210031                      LD      HL,FIXBUF
26CF 0600                        LD      B,0
26D1 CDEF2C                      CALL    $OPEN
26D4 210048                      LD      HL,PGMDATA      ;Pt HL to highest byte avail
26D7 2B                          DEC     HL              ;  for fix data
26D8 010034                      LD      BC,FIXDATA      ;Start of /FIX data storage
26DB CD072D              RDFIX1  CALL    $GET1           ;Get a char fm /FIX file
26DE 200F                        JR      NZ,RDFIX2       ;Jump on error
26E0 E67F                        AND     7FH             ;Strip bit 7
26E2 2810                        JR      Z,RDFIX3        ;Take 0 as EOF also
26E4 02                          LD      (BC),A          ;Save fix data char
26E5 03                          INC     BC              ;Advance buffer
26E6 E5                          PUSH    HL              ;Save HL tempy
26E7 ED42                        SBC     HL,BC           ;Room in fixdata buffer?
26E9 E1                          POP     HL
26EA DA4D2D                      JP      C,TOOBIG        ;Abort if patch data too large
26ED 18EC                        JR      RDFIX1          ;  else loop til EOF
                         ;
26EF FE1C                RDFIX2  CP      1CH             ;End of file?
26F1 C2322D                      JP      NZ,IOERR        ;Abort if not
26F4 3E03                RDFIX3  LD      A,ETX           ;Mark the end of the fix data
```

```
26F6 Ø2                     LD      (BC),A
                    ;
                    ;       Start patching the target file
                    ;
26F7 21ØØ34         DOFIX   LD      HL,FIXDATA      ;Pt to start of fix data
                    ;
26FA E5             DOFIX1  PUSH    HL
26FB 21882E                 LD      HL,RDGINP$      ;"reading input...
26FE CD1F2D                 CALL    $DSPLY
27Ø1 E1                     POP     HL
27Ø2 226B2D                 LD      (SETMSG+1),HL   ;Used if error in line
27Ø5 3EØØ                   LD      A,$-$
27Ø6                PASS2   EQU     $-1             ;Zero if 1st pass thru data
27Ø7 B7                     OR      A
27Ø8 7E                     LD      A,(HL)          ;P/U a character
27Ø9 CA142C                 JP      Z,PASS1         ;Go if 1st pass
27ØC 7E                     LD      A,(HL)
27ØD FEØ3                   CP      ETX             ;End of patch?
27ØF 285C                   JR      Z,PCHDUN
2711 FE2E                   CP      '.'             ;Comment?
2713 CAC527                 JP      Z,COMMENT
2716 CBAF                   RES     5,A             ;Make upper case
2718 FE46                   CP      'F'             ;FIND line?
271A CAC527                 JP      Z,COMMENT       ;Skip on 2nd pass or if O=N
271D FE44                   CP      'D'             ;Start of D line?
271F CAD727                 JP      Z,DVERB
2722 FE59                   CP      'Y'             ;Yank previous patch?
2724 CA4228                 JP      Z,YANK
2727 FE4C                   CP      'L'             ;Library overlay?
2729 CAØØ29                 JP      Z,LVERB
272C FE52                   CP      'R'             ;Remove parm ?
272E CA3A28                 JP      Z,REMOVE
2731 FE4F                   CP      'O'             ;O parm ?
2733 CAD128                 JP      Z,OVERB
2736 FE58                   CP      'X'             ;Start of X line?
2738 C2612D                 JP      NZ,PCHERR       ;Error if none of above
                    ;
                    ;       Verb = 'X' -> patch by hex load address
                    ;
273B 117F2D                 LD      DE,PGMDCB       ;Rewind the program to Ø
273E Ø1ØØØØ                 LD      BC,Ø            ;Use POSN so EOF
2741 CDF52C                 CALL    $POSN           ;  is not changed
2744 CD4D29                 CALL    POSFIL          ;Posn to end of prgfile
                    ;
2747 F5                     PUSH    AF              ;Save regs fm display routine
2748 E5                     PUSH    HL
2749 D5                     PUSH    DE
274A 21AD2E                 LD      HL,INSPCH$      ;"installing patch...
274D CD1F2D                 CALL    $DSPLY
275Ø D1                     POP     DE
2751 E1                     POP     HL
2752 F1                     POP     AF
                    ;
2753 FEØ2                   CP      2               ;Be sure type byte = 2
2755 C2452D                 JP      NZ,FILERR       ;Load file format error
2758 3EØ1                   LD      A,1             ;Tempy set LRL to 1
275A 32882D                 LD      (PGMDCB+9),A    ;  & backspace the file
275D CDFB2C                 CALL    $BKSP           ;  to overwrite old xfer addr
276Ø AF                     XOR     A               ;Reset LRL to 256
2761 32882D                 LD      (PGMDCB+9),A
                    ;
```

```
               02320 ;        Install the X patch at the end of the prgfile
               02330 ;
2764 CD8229    02340          CALL     STUFNM          ;Generate the patch
2767 7E        02350          LD       A,(HL)          ;HL => ending posn in fix data
2768 FE03      02360          CP       ETX             ;Did it go til the end?
276A C2612D    02370          JP       NZ,PCHERR       ;"Patch format error...
               02380 ;
               02390 ;        Patch/operation complete - close the file
               02400 ;
276D 3E0D      02410 PCHDUN   LD       A,CR            ;Move cursor to next line
276F CD2B2D    02420          CALL     $DSP
2772 117F2D    02430          LD       DE,PGMDCB       ;Close the program file
2775           02440          @@CLOSE
2775 3E3C      00017          LD       A,60
2777 EF        00018          RST      40
2778 C2322D    02450          JP       NZ,IOERR
277B 21EE2F    02460          LD       HL,YANKMSG      ;Set up in case Yank was done
277E 3A732D    02470          LD       A,(YNKFLG)      ;Was it a Yank?
2781 B7        02480          OR       A
2782 2024      02490          JR       NZ,EXLOG        ;Yes, log out
2784 21962F    02500          LD       HL,SUCCES$      ;"function completed.."
2787           02510          @@LOGOT
               00019          IFEQ     00H,1
               00020          LD       HL,
               00021          ENDIF
2787 3E0C      00022          LD       A,12
2789 EF        00023          RST      40
278A 2A742D    02520          LD       HL,(LINCNT)     ;P/u # of D & X lines
278D 7C        02530          LD       A,H
278E B5        02540          OR       L               ;Any?
278F 2814      02550          JR       Z,NOCHG         ;No D or X verbs
2791 E5        02560          PUSH     HL              ;Save line count
2792 110100    02570          LD       DE,1            ;Exactly 1 line?
2795 ED52      02580          SBC      HL,DE
2797 E1        02590          POP      HL
2798 2005      02600          JR       NZ,NTONE        ;Go if more than 1
279A 3E20      02610          LD       A,' '           ;  else remove "s" from message
279C 32C12F    02620          LD       (PLURAL),A
279F 11B12F    02630 NTONE    LD       DE,LINMSG$      ;Put line count into message
27A2           02640          @@HEXDEC                 ;  as decimal ASCII
27A2 3E61      00024          LD       A,97
27A4 EF        00025          RST      40
27A5 21B12F    02650 NOCHG    LD       HL,LINMSG$
27A8           02660 EXLOG    @@LOGOT                  ;Show how many lines done
               00026          IFEQ     00H,1
               00027          LD       HL,
               00028          ENDIF
27A8 3E0C      00029          LD       A,12
27AA EF        00030          RST      40
               02670 ;
27AB 210000    02680          LD       HL,0            ;Init no error
27AE E5        02690 $QUIT    PUSH     HL
27AF 217F2D    02700          LD       HL,PGMDCB
27B2 CB7E      02710          BIT      7,(HL)          ;Was file left open?
27B4 EB        02720          EX       DE,HL           ;DE=>DCB possible close
27B5 C4D62C    02730          CALL     NZ,FLOPN        ;Warn user
27B8 3E0E      02740          LD       A,14            ;Cursor on
27BA CD2B2D    02750          CALL     $DSP
27BD E1        02760          POP      HL
27BE 310000    02770          LD       SP,$-$          ;P/u original stack
27BF           02780 STACK    EQU      $-2
```

```
27C1          Ø279Ø          @@CKBRKC                ;Clear break
27C1 3E6A     ØØØ31          LD      A,1Ø6
27C3 EF       ØØØ32          RST     4Ø
27C4 C9       Ø28ØØ          RET                     ;Done with the patching
              Ø281Ø ;
              Ø282Ø ;        Verb = '.' => comment line
              Ø283Ø ;        HL = start of line in fix data
              Ø284Ø ;        Bypass all chars until a terminator is found
              Ø285Ø ;
27C5 7E       Ø286Ø COMMENT  LD      A,(HL)          ;Look for some terminator
27C6 FEØ3     Ø287Ø          CP      ETX             ;End of the fix data?
27C8 CAFA26   Ø288Ø          JP      Z,DOFIX1        ;Back if so
27CB 23       Ø289Ø          INC     HL              ;  else bump buffer ptr
27CC FE3B     Ø29ØØ          CP      ';'             ;Logical EOL?
27CE 28Ø4     Ø291Ø          JR      Z,EOL1          ;Back if so
27DØ FEØD     Ø292Ø          CP      CR              ;Physical EOL?
27D2 2ØF1     Ø293Ø          JR      NZ,COMMENT      ;Do next char if not
27D4 C3FA26   Ø294Ø EOL1     JP      DOFIX1          ;Back to the caller
              Ø295Ø ;
              Ø296Ø ;        Verb = 'D' -> disk record patch
              Ø297Ø ;
27D7 CDCC2C   Ø298Ø DVERB    CALL    CNTLIN          ;Bump line counter
27DA CDE327   Ø299Ø          CALL    DPOSN           ;Posn prgfile to Drr,bb
27DD CDØA28   Ø3ØØØ          CALL    DLINE           ;Put or check the patch line
              Ø3Ø1Ø                                  ;  depending on which pass
27EØ C3FA26   Ø3Ø2Ø          JP      DOFIX1          ;Do next line
              Ø3Ø3Ø ;
27E3 23       Ø3Ø4Ø DPOSN    INC     HL              ;Bump fix data buffer ptr
27E4 CD922A   Ø3Ø5Ø          CALL    PRSFIX          ;Get char or hex pair
27E7 Ø6ØØ     Ø3Ø6Ø          LD      B,Ø             ;Put disk record #
27E9 4F       Ø3Ø7Ø          LD      C,A             ;  into BC
27EA 7E       Ø3Ø8Ø          LD      A,(HL)          ;If no comma, then
27EB FE2C     Ø3Ø9Ø          CP      ','             ; get 3rd & 4th digits
27ED 28Ø4     Ø31ØØ          JR      Z,DVERB1        ;  in case user put in
27EF CD922A   Ø311Ø          CALL    PRSFIX          ;  a 4 byte record #
27F2 4F       Ø312Ø          LD      C,A
27F3 117F2D   Ø313Ø DVERB1   LD      DE,PGMDCB       ;Position file to record
27F6 CDF52C   Ø314Ø          CALL    $POSN
27F9 7E       Ø315Ø          LD      A,(HL)          ;Check for ',' separator
27FA FE2C     Ø316Ø          CP      ','             ;  between record and offset
27FC C2612D   Ø317Ø          JP      NZ,PCHERR       ;Abort if not found
27FF 23       Ø318Ø          INC     HL              ;Pt to offset bytes
28ØØ CD252D   Ø319Ø          CALL    $READ           ;Read the sector
28Ø3 CD922A   Ø32ØØ          CALL    PRSFIX          ;Make offset binary in A
28Ø6 32842D   Ø321Ø          LD      (PGMDCB+5),A    ;Set byte offset in FCB
28Ø9 C9       Ø322Ø          RET
              Ø323Ø ;
              Ø324Ø ;        Dual purpose routine that checks a Drr,bb line
              Ø325Ø ;        or installs it into the program file
              Ø326Ø ;
28ØA 7E       Ø327Ø DLINE    LD      A,(HL)          ;Next byte in line must
28ØB FE3D     Ø328Ø          CP      '='             ;  be '='
28ØD C2612D   Ø329Ø          JP      NZ,PCHERR       ;Abort if missing
281Ø 23       Ø33ØØ DVERB2   INC     HL              ;Pt to start of patch data
2811 CD962A   Ø331Ø DVERB3   CALL    PRSFX1          ;Get patch byte as binary in A
2814 CDB82C   Ø332Ø          CALL    PUTORCHK        ;Either write it or check it
2817 7E       Ø333Ø          LD      A,(HL)          ;P/u next char
2818 FEØD     Ø334Ø          CP      CR              ;Go on CR
281A 2811     Ø335Ø          JR      Z,DVERB4A
281C FE3B     Ø336Ø          CP      ';'             ;End of logical line?
281E 28ØC     Ø337Ø          JR      Z,DVERB4
```

```
282Ø FE22                CP      '"'             ;Closing dbl-quote?
2822 28Ø8                JR      Z,DVERB4
2824 3AAB2A              LD      A,(STRFLG+1)    ;If in quote string,
2827 B7                  OR      A               ;  do not bump HL past
2828 28E6                JR      Z,DVERB2        ;  the non-existant space
282A 18E5                JR      DVERB3          ;No special, do next byte
                 ;
282C 7E          DVERB4  LD      A,(HL)          ;Ignore rest of line
282D 23          DVERB4A INC     HL
282E FEØD                CP      CR
283Ø 2ØFA                JR      NZ,DVERB4       ;Loop til physical EOL
2832 3AØ627              LD      A,(PASS2)       ;Patching or checking?
2835 B7                  OR      A               ;If patching, need to
2836 C4Ø12D              CALL    NZ,$RWRIT       ;  re-write the sector
2839 C9                  RET                     ;Done w/line
                 ;
                 ;       Verb = 'R' -> set flag to yank D patch
                 ;       This routine is needed to check the R parm
                 ;        when doing a CLP, in case the parm was entered
                 ;        after the fix data
                 ;
283A 3EFF        REMOVE  LD      A,-1            ;Set Reomve parm true and
283C 32472C              LD      (RPARM),A       ;  then ignore all until the
283F C3C527              JP      COMMENT         ;  next logical line
                 ;
                 ;       Verb = 'Y' -> yanks patch with same name
                 ;
2842 7E          YANK    LD      A,(HL)          ;Ignore all chars until
2843 23                  INC     HL              ;  the physical EOL
2844 FEØD                CP      CR
2846 2ØFA                JR      NZ,YANK
                 ;
2848 E5                  PUSH    HL              ;Save fix data posn
2849 21DC2E              LD      HL,YNKPCH$      ;"yanking patch...
284C CD1F2D              CALL    $DSPLY
284F Ø1ØØØØ              LD      BC,Ø            ;Rewind the file
2852 117F2D              LD      DE,PGMDCB
2855 CDF52C              CALL    $POSN
2858 CDØ72D      YANK1   CALL    $GET1           ;Get a "type" byte
285B C2C628              JP      NZ,YANK9        ;If error, ck for EOF
285E FEØ7                CP      7               ;Found a patch?
286Ø 281B                JR      Z,YANK4         ;If so, check name
2862 326A28              LD      (TYPCOD+1),A    ;Stuff type for testing
2865 CDØB2D              CALL    $GET            ;Get a block length
2868 47                  LD      B,A             ;Set loop counter
2869 3EØØ        TYPCOD  LD      A,Ø             ;Test type
286B 3D                  DEC     A               ;Ck for type 1 (code block)
286C 2ØØ8                JR      NZ,YANK2        ;Length ok if not
                 ;
                 ;       Adjust length for 255 & 256 byte code blocks
                 ;
286E CDØB2D              CALL    $GET            ;Read 1st two bytes
2871 Ø5                  DEC     B               ;  in case the block was
2872 CDØB2D              CALL    $GET            ;  255+2 or 256+2
2875 Ø5                  DEC     B
2876 CDØB2D      YANK2   CALL    $GET            ;Read rest of block
2879 1ØFB        YANK3   DJNZ    YANK2
287B 18DB                JR      YANK1
                 ;
                 ;       Found patch code area, is this the one?
                 ;
```

```
287D CDØB2D    Ø399Ø YANK4    CALL     $GET             ;Get name len fm file
288Ø 47        Ø4ØØØ          LD       B,A              ;Save len in B
2881 3AØ83Ø    Ø4Ø1Ø          LD       A,(NAMLEN$)      ;P/u fix file name length
2884 B8        Ø4Ø2Ø          CP       B                ;If no match, not fix
2885 2ØEF      Ø4Ø3Ø          JR       NZ,YANK2         ;  to Yank
2887 21Ø93Ø    Ø4Ø4Ø          LD       HL,NAMFIX$       ;Pt to yank file name
288A CDØ72D    Ø4Ø5Ø YANK5    CALL     $GET1            ;Ck for match of yank
288D C27628    Ø4Ø6Ø          JP       NZ,YANK2         ;  file name with prog
289Ø BE        Ø4Ø7Ø          CP       (HL)             ;  patch name
2891 23        Ø4Ø8Ø          INC      HL
2892 2ØE5      Ø4Ø9Ø          JR       NZ,YANK3         ;Back if no match
2894 1ØF4      Ø41ØØ          DJNZ     YANK5
               Ø411Ø ;
               Ø412Ø ;        Found this fix patch - let's yank it
               Ø413Ø ;
2896 CDØB2D    Ø414Ø YANK6    CALL     $GET             ;Get type code
2899 FEØ1      Ø415Ø          CP       1                ;Ignore block if
289B C2BD28    Ø416Ø          JP       NZ,YANK8         ;  type <> 1 (code block)
289E 3EØ1      Ø417Ø          LD       A,1              ;Set LRL=1 & backspace
28AØ 32882D    Ø418Ø          LD       (PGMDCB+9),A     ;  to overwrite the type byte
28A3 CDFB2C    Ø419Ø          CALL     $BKSP
28A6 AF        Ø42ØØ          XOR      A                ;Set LRL back to 256
28A7 32882D    Ø421Ø          LD       (PGMDCB+9),A
28AA 3E1Ø      Ø422Ø          LD       A,1ØH            ;Change type=1 to =16
28AC CD112D    Ø423Ø          CALL     $PUT             ;  and write to prgfile
28AF CDØ12D    Ø424Ø          CALL     $RWRIT           ;Force re-write
28B2 CDØB2D    Ø425Ø          CALL     $GET             ;Get length byte
28B5 47        Ø426Ø          LD       B,A              ;  of patch code block
28B6 CDØB2D    Ø427Ø YANK7    CALL     $GET
28B9 1ØFB      Ø428Ø          DJNZ     YANK7            ;Posn past the code block
28BB 18D9      Ø429Ø          JR       YANK6            ;Loop through patch blocks
               Ø43ØØ ;
28BD E1        Ø431Ø YANK8    POP      HL               ;Not type 1, done with yank
28BE 3EFF      Ø432Ø          LD       A,ØFFH           ;Set Yank flag for
28CØ 32732D    Ø433Ø          LD       (YNKFLG),A       ;  exit message dsply
28C3 C36D27    Ø434Ø          JP       PCHDUN
               Ø435Ø ;
28C6 FE1C      Ø436Ø YANK9    CP       1CH              ;Got $GET error, was EOF?
28C8 C2322D    Ø437Ø          JP       NZ,IOERR         ;Abort if not, else
28CB 21F72E    Ø438Ø          LD       HL,NOYANK$       ;  "can't yank, not in file
28CE C3542D    Ø439Ø          JP       ERREXIT
               Ø44ØØ ;
               Ø441Ø ;        Verb = 'O' -> turn FIND on/off
               Ø442Ø ;        Check special O parameter, determine ON or OFF
               Ø443Ø ;
28D1 23        Ø444Ø OVERB    INC      HL               ;Move past O
28D2 7E        Ø445Ø          LD       A,(HL)
28D3 FE3D      Ø446Ø          CP       '='              ;Next char must be '='
28D5 2Ø1D      Ø447Ø          JR       NZ,WHATIS        ;  or is an error
28D7 23        Ø448Ø          INC      HL               ;Bypass the '='
28D8 7E        Ø449Ø          LD       A,(HL)
28D9 FEØD      Ø45ØØ          CP       CR               ;Was it CR or ')'?
28DB 281B      Ø451Ø          JR       Z,OISOFF         ;O=<enter> is OFF
28DD CBAF      Ø452Ø          RES      5,A              ;Make Upper case
28DF FE4E      Ø453Ø          CP       'N'
28E1 2815      Ø454Ø          JR       Z,OISOFF         ;O=N,NO etc.
28E3 FE59      Ø455Ø          CP       'Y'              ;Y=yes
28E5 281Ø      Ø456Ø          JR       Z,OISON
28E7 FE4F      Ø457Ø          CP       'O'
28E9 2ØØ9      Ø458Ø          JR       NZ,WHATIS        ;Not Y/N/ON/OFF!
28EB CDB32C    Ø459Ø          CALL     GETNXT           ;Get nxt, already UC
```

```
28EE FE46                 CP      'F'
28FØ 28Ø6                 JR      Z,OISOFF        ;OFF
28F2 FE4E                 CP      'N'
28F4 C2612D       WHATIS  JP      NZ,PCHERR       ;Quit if no acceptable flag
                  ;
28F7 3E           OISON   DB      3EH             ;LD A,ØAFH
28F8 AF           OISOFF  XOR     A
28F9 32412C               LD      (OPARM),A       ;Set parm on or off
28FC 2B                   DEC     HL
28FD C3C527               JP      COMMENT         ;Ignore rest til logical EOL
                  ;
                  ;       Verb = 'L' -> indicate patch to library file
                  ;
29ØØ 23           LVERB   INC     HL              ;Bypass the 'L'
29Ø1 CD922A               CALL    PRSFIX          ;Get a hex digit pair
29Ø4 4F                   LD      C,A             ;Stuff for later
29Ø5 328F2B               LD      (OVRLY+1),A
29Ø8 7E                   LD      A,(HL)          ;Ck for end of line
29Ø9 23                   INC     HL
29ØA FEØD                 CP      CR
29ØC C2612D               JP      NZ,PCHERR       ;Error if not
29ØF CDE92A               CALL    FISAM           ;Get isam overlay ptrs
2912 F5                   PUSH    AF              ;Save byte offset
2913 3A8Ø2D               LD      A,(PGMDCB+1)
2916 CBBF                 RES     7,A             ;Sector operations only
2918 328Ø2D               LD      (PGMDCB+1),A
291B 117F2D               LD      DE,PGMDCB       ;Position the file to
291E CDF52C               CALL    $POSN           ;Overlay requested
2921 CD252D               CALL    $READ           ;Read in the sector
2924 F1                   POP     AF
2925 32842D               LD      (PGMDCB+5),A    ;Stuff byte offset in FCB
2928 CD4D29               CALL    POSFIL          ;Adv "positioning...
292B FEØ4                 CP      4               ;End of ISAM overlay?
292D C2452D               JP      NZ,FILERR       ;If not, "load format er.
293Ø 3EØ1                 LD      A,1             ;Set LRL=1
2932 32882D               LD      (PGMDCB+9),A
2935 CDFB2C               CALL    $BKSP           ;Backspace over the 4
2938 AF                   XOR     A               ;Now set LRL back to 256
2939 32882D               LD      (PGMDCB+9),A
293C CD8229               CALL    STUFNM          ;Do the patch
293F E5                   PUSH    HL
294Ø 21C12E               LD      HL,BLDMAP$      ;"rebuilding library map.
2943 CD1F2D               CALL    $DSPLY
2946 CD3A2B               CALL    RPRMAP          ;Rebuild the map
2949 E1                   POP     HL
294A C3FA26               JP      DOFIX1          ;Loop
                  ;
                  ;       Include the rest of Patch/Cmd
                  ;
294D              *GET    PATCHA:3
                  ;PATCHA/ASM - Continuation of Patch Program
                  ;
                  ;       Routine to position to file's end
                  ;
294D E5           POSFIL  PUSH    HL              ;Save fm display call
294E D5                   PUSH    DE
294F 216F2E               LD      HL,POSLD$       ;"positioning ...
2952 CD1F2D               CALL    $DSPLY
2955 D1                   POP     DE
2956 E1                   POP     HL
                  ;
```

```
2957 CD0B2D     04060 POSFIL1 CALL    $GET             ;Get a type byte
295A FE20       04070         CP      20H              ;X'20' & up are illegal
295C D2452D     04080         JP      NC,FILERR
295F FE02       04090         CP      2                ;Transfer address?
2961 C8         04100         RET     Z
2962 FE03       04110         CP      3                ;Not really used in
2964 C8         04120         RET     Z                ;  a file, yet...
2965 FE04       04130         CP      4                ;End of ISAM member?
2967 C8         04140         RET     Z
2968 FE0A       04150         CP      0AH              ;End of ISAM directory?
296A C8         04160         RET     Z
296B 4F         04170         LD      C,A              ;Save type byte
296C CD0B2D     04180         CALL    $GET             ;Get block length
296F 47         04190         LD      B,A              ;Save it for countdown
2970 0D         04200         DEC     C                ;Was type = 1 ?
2971 2008       04210         JR      NZ,POSFIL2       ;Jump if not
2973 CD0B2D     04220         CALL    $GET             ;Read off the load addr
2976 05         04230         DEC     B                ;Adjust length for each
2977 CD0B2D     04240         CALL    $GET
297A 05         04250         DEC     B
297B CD0B2D     04260 POSFIL2 CALL    $GET             ;Read the block
297E 10FB       04270         DJNZ    POSFIL2
2980 18D5       04280         JR      POSFIL1          ;Loop to next type code
                04290 ;
                04300 ;       Routine to put the patch name header block into the
                04310 ;        prg data buffer and then position to the next X'' line
                04320 ;
2982 E5         04330 STUFNM  PUSH    HL               ;Save posn in fix data
2983 21992E     04340         LD      HL,GENPCH$       ;"generating patch...
2986 CD1F2D     04350         CALL    $DSPLY
2989 110048     04360         LD      DE,PGMDATA
298C 210830     04370         LD      HL,NAMLEN$       ;Pt to fix name field
298F 7E         04380         LD      A,(HL)
2990 B7         04390         OR      A
2991 280C       04400         JR      Z,STUFNM2        ;Go if no name len
2993 3E07       04410         LD      A,7              ;Set fix patch type
2995 12         04420         LD      (DE),A
2996 13         04430         INC     DE
2997 46         04440         LD      B,(HL)           ;Set header length
2998 04         04450         INC     B                ;Bump to write length
2999 7E         04460 STUFNM1 LD      A,(HL)           ;P/u name byte
299A 23         04470         INC     HL
299B 12         04480         LD      (DE),A           ;Put in output buffer
299C 13         04490         INC     DE
299D 10FA       04500         DJNZ    STUFNM1          ;Loop for namelen
                04510 ;
299F E1         04520 STUFNM2 POP     HL               ;Recover posn in fix data
29A0 226B2D     04530 STUFNM3 LD      (SETMSG+1),HL    ;Start of this line
29A3 7E         04540         LD      A,(HL)
29A4 FE03       04550         CP      ETX              ;End of fix data?
29A6 CA2F2A     04560         JP      Z,RIPPLE
29A9 23         04570         INC     HL
29AA FE2E       04580         CP      '.'              ;Comment?
29AC 2809       04590         JR      Z,STUFNM4
29AE CBAF       04600         RES     5,A              ;In case lower case
29B0 FE58       04610         CP      'X'              ;Start of code line?
29B2 2810       04620         JR      Z,DOXVB          ;Go if so
29B4 C3612D     04630         JP      PCHERR           ;"patch input format err
                04640 ;
29B7 7E         04650 STUFNM4 LD      A,(HL)           ;In a comment, loop until
29B8 23         04660         INC     HL               ;  end of line
```

```
29B9 FEØ3                   CP      ETX             ;End of patch code?
29BB CA612D                 JP      Z,PCHERR        ;Abort if so
29BE FEØD                   CP      CR              ;EOL?
29CØ 2ØF5                   JR      NZ,STUFNM4      ;Loop if not
29C2 18DC                   JR      STUFNM3
                   ;
                   ;        Do the 'X' verb patch
                   ;        HL => Fix data buffer
                   ;        DE => Program data buffer
                   ;
29C4 CDCC2C        DOXVB    CALL    CNTLIN          ;Count installed lines
29C7 3EØ1                   LD      A,1             ;Show type 1 (code block)
29C9 12                     LD      (DE),A          ;Put in output buffer
29CA 13                     INC     DE
29CB D5                     PUSH    DE              ;Save ptr to length
29CC 13                     INC     DE
29CD 7E                     LD      A,(HL)          ;Should be "'"
29CE 23                     INC     HL              ;  around address (X'nnnn')
29CF FE27                   CP      27H
29D1 C2612D                 JP      NZ,PCHERR       ;Error if not
29D4 CD922A                 CALL    PRSFIX          ;P/u hex digit pair
29D7 47                     LD      B,A             ;Save hi-order address
29D8 CD922A                 CALL    PRSFIX          ;P/u hex digit pair
29DB 12                     LD      (DE),A          ;Stuff lo-order address
29DC 13                     INC     DE
29DD 78                     LD      A,B
29DE 12                     LD      (DE),A          ;Stuff hi-order address
29DF 13                     INC     DE
29EØ 7E                     LD      A,(HL)          ;Syntax requires "=" or
29E1 FE3D                   CP      '='             ;  "'" next
29E3 28Ø6                   JR      Z,DOXVB1
29E5 FE27                   CP      27H             ;Bypass optional clsng '
29E7 C2612D                 JP      NZ,PCHERR       ;Error if not ',=
29EA 23                     INC     HL
29EB 23            DOXVB1   INC     HL              ;Bypass the '='
29EC Ø6Ø2                   LD      B,2             ;Len of bytes already stuffed
29EE 7E            DOXVB2   LD      A,(HL)          ;Get char of fix data
29EF FE22                   CP      '"'             ;ASCII string?
29F1 281F                   JR      Z,DOXVB5        ;Go process if so
29F3 7E            DOXVB3   LD      A,(HL)          ;P/u line byte
29F4 23                     INC     HL
29F5 FE3B                   CP      ';'             ;Logical end?
29F7 2811                   JR      Z,DOXVB4        ;Ignore trailing
29F9 FEØD                   CP      CR              ;End of line?
29FB 282C                   JR      Z,DOXVB6
29FD FE2Ø                   CP      2ØH
29FF 28ED                   JR      Z,DOXVB2        ;Ignore spaces
2AØ1 2B                     DEC     HL              ;Back up, its a byte
2AØ2 CD962A                 CALL    PRSFX1          ;Get the hex digit pair
2AØ5 12                     LD      (DE),A          ;Stuff into code buffer
2AØ6 13                     INC     DE
2AØ7 Ø4                     INC     B               ;Bump block length
2AØ8 18E9                   JR      DOXVB3
                   ;
                   ;        Bypass until end of line
                   ;
2AØA 7E            DOXVB4   LD      A,(HL)          ;P/u the character
2AØB 23                     INC     HL
2AØC FEØD                   CP      CR              ;End of line?
2AØE 2ØFA                   JR      NZ,DOXVB4
2A1Ø 1817                   JR      DOXVB6
```

```
                         ;
                         ;       Fix has double quote string
                         ;
2A12 23                  DOXVB5  INC     HL
2A13 7E                          LD      A,(HL)          ;Get next char
2A14 FEØ3                        CP      ETX             ;End of fix data?
2A16 CA612D                      JP      Z,PCHERR        ;Can't end w/o some EOL
2A19 23                          INC     HL
2A1A FEØD                        CP      CR              ;End of line?
2A1C CA292A                      JP      Z,DOXVB6        ;Valid end
2A1F FE22                        CP      '"'             ;Closing quote?
2A21 28DØ                        JR      Z,DOXVB3        ;Go for more
2A23 2B                          DEC     HL
2A24 12                          LD      (DE),A          ;Stuff the char
2A25 13                          INC     DE
2A26 Ø4                          INC     B               ;Bump counter
2A27 18E9                        JR      DOXVB5          ;Loop until end or "
                         ;
                         ;       Found valid end - update length
                         ;
2A29 E3                  DOXVB6  EX      (SP),HL         ;Grab length pointer
2A2A 7Ø                          LD      (HL),B          ;Stuff the length
2A2B E1                          POP     HL
2A2C C3AØ29                      JP      STUFNM3         ;Go for more lines
                         ;
                         ;       Got to the end of the fix input
                         ;
2A2F E5                  RIPPLE  PUSH    HL
2A3Ø EB                          EX      DE,HL           ;Last patch byte to HL
2A31 11ØØ48                      LD      DE,PGMDATA      ;Pt to patch code buffer
2A34 AF                          XOR     A
2A35 ED52                        SBC     HL,DE           ;Calc length of fixup
2A37 22C62B                      LD      (RPRMAP9+1),HL  ;Stuff for later
2A3A 21AD2E                      LD      HL,INSPCH$      ;"installing patch
2A3D CD1F2D                      CALL    $DSPLY
2A4Ø 217F2D                      LD      HL,PGMDCB       ;Move prog into fix
2A43 11AØ2D                      LD      DE,FIXDCB       ;  file control block
2A46 Ø12ØØØ                      LD      BC,32           ;  for output use
2A49 EDBØ                        LDIR
2A4B 21ØØ32                      LD      HL,LIBBUF       ;Set the i/o buffer
2A4E 22A32D                      LD      (FIXDCB+3),HL
2A51 11AØ2D                      LD      DE,FIXDCB       ;Reread the last program
2A54                             @@RREAD                 ;  sector
2A54 3E45                        LD      A,69
2A56 EF                          RST     4Ø
2A57 C2322D                      JP      NZ,IOERR        ;Quit on read error
                         ;
                         ;       Now ripple the file down while stuffing bytes
                         ;
2A5A 21ØØ48                      LD      HL,PGMDATA      ;Beginning of "fixed" code
2A5D 11AØ2D              RIPPL1  LD      DE,FIXDCB       ;Get prog byte
2A6Ø CDØ72D                      CALL    $GET1
2A63 C27A2A                      JP      NZ,RIPPL2
2A66 E5                          PUSH    HL              ;Save buffer ptr & byte
2A67 F5                          PUSH    AF
2A68 117F2D                      LD      DE,PGMDCB       ;Use the output fcb
2A6B 7E                          LD      A,(HL)          ;P/u byte from fixbuf
2A6C CD112D                      CALL    $PUT            ;Put to disk
2A6F ED4BC62B                    LD      BC,(RPRMAP9+1)  ;Pt to patch length
2A73 Ø9                          ADD     HL,BC           ;Pt past patch code
2A74 F1                          POP     AF              ;Recover prog byte
```

```
2A75 77                     LD      (HL),A          ;  & stuff after fix code
2A76 E1                     POP     HL              ;Rcvr buf ptr
2A77 23                     INC     HL              ;Bump & loop
2A78 18E3                   JR      RIPPL1
                  ;
2A7A FE1C         RIPPL2    CP      1CH             ;Got to end of file?
2A7C C2452D                 JP      NZ,FILERR       ;Quit on any other error
2A7F 117F2D                 LD      DE,PGMDCB
2A82 ED4BC62B               LD      BC,(RPRMAP9+1)  ;Get length of patch
2A86 7E           RIPPL3    LD      A,(HL)          ;Put rest of program
2A87 23                     INC     HL              ;  (ie the bytes = to
2A88 CD112D                 CALL    $PUT            ;  length of patch code)
2A8B ØB                     DEC     BC              ;Do until len left = Ø
2A8C 78                     LD      A,B
2A8D B1                     OR      C
2A8E 2ØF6                   JR      NZ,RIPPL3
2A9Ø E1                     POP     HL
2A91 C9                     RET
                  ;
                  ;         Routine to read & convert fix code values
                  ;
2A92 AF           PRSFIX    XOR     A               ;Entry to clear
2A93 32AB2A                 LD      (STRFLG+1),A    ;  STRING check
2A96 7E           PRSFX1    LD      A,(HL)          ;P/u patch char
2A97 FEØ3                   CP      ETX             ;End of text?
2A99 CA612D                 JP      Z,PCHERR        ;Error if so
2A9C FE22                   CP      '"'             ;String?
2A9E 2ØØA                   JR      NZ,STRFLG
2AAØ 32AB2A                 LD      (STRFLG+1),A    ;Stuff string indicator
2AA3 23                     INC     HL
2AA4 7E                     LD      A,(HL)          ;P/u char
2AA5 FEØ3                   CP      ETX             ;End again?
2AA7 CA612D                 JP      Z,PCHERR
2AAA 3EØØ         STRFLG    LD      A,Ø             ;Test string flag
2AAC B7                     OR      A
2AAD 7E                     LD      A,(HL)          ;P/u char again
2AAE 23                     INC     HL              ;Bump pointer
2AAF CØ                     RET     NZ              ;Ret if '"' was prev char
2ABØ CDD72A                 CALL    CVTBIN          ;Convert hex digit to bin
2AB3 4F                     LD      C,A             ;Save value
2AB4 7E                     LD      A,(HL)          ;P/u next digit
2AB5 23                     INC     HL
2AB6 FEØ3                   CP      ETX             ;Backup pointer and ret
2AB8 CA612D                 JP      Z,PCHERR        ;  if next char is not hex
2ABB FE3Ø                   CP      'Ø'             ;  else pack it into regC
2ABD 3815                   JR      C,PRSFX3        ;  & place in reg A
2ABF FE3A                   CP      '9'+1
2AC1 38Ø4                   JR      C,PRSFX2
2AC3 FE41                   CP      'A'
2AC5 38ØD                   JR      C,PRSFX3
2AC7 CBØ1         PRSFX2    RLC     C               ;Assume digit, move
2AC9 CBØ1                   RLC     C               ;  over a nybble
2ACB CBØ1                   RLC     C
2ACD CBØ1                   RLC     C
2ACF CDD72A                 CALL    CVTBIN          ;Get hex digit
2AD2 B1                     OR      C               ;Merge hi-order nybble
2AD3 C9                     RET
2AD4 79           PRSFX3    LD      A,C             ;Non-hex char,
2AD5 2B                     DEC     HL              ;  rcvr & exit
2AD6 C9                     RET
                  ;
```

```
                ;       Routine to convert hex digit to binary
                ;
2AD7 D63Ø       CVTBIN  SUB     3ØH             ;1st adjustment to binary
2AD9 DA5D2D             JP      C,NONHEX        ;Quit if too low
2ADC FEØA               CP      1Ø              ;Ø-9 range?
2ADE D8                 RET     C               ;Back if so
2ADF CBAF               RES     5,A             ;In case lower case
2AE1 D6Ø7               SUB     7
2AE3 FE1Ø               CP      16              ;Less than F+1?
2AE5 D8                 RET     C               ;Ok if so
2AE6 C35D2D             JP      NONHEX          ;  else abort
                ;
                ;       Routine to find ISAM member pointer in map table
                ;
2AE9 117F2D     FISAM   LD      DE,PGMDCB
2AEC CDØ72D     FISAM1  CALL    $GET1           ;Get a type byte
2AEF 28Ø8               JR      Z,FISAM1A       ;Go on no error
2AF1 FE1C               CP      1CH             ;EOF?
2AF3 CA412D             JP      Z,LIBERR        ;Invalid library format
2AF6 C3322D             JP      IOERR           ;  else I/O error
2AF9 FEØ8       FISAM1A CP      8               ;Start of map table?
2AFB 282Ø               JR      Z,FISAM3
2AFD FEØA               CP      ØAH             ;End of map table?
2AFF CA3D2D             JP      Z,NOVRLY        ;Should not be end
2BØ2 C5                 PUSH    BC
2BØ3 4F                 LD      C,A             ;Save TYPE
2BØ4 CDØB2D             CALL    $GET            ;Get block length
2BØ7 47                 LD      B,A             ;Set counter & read
2BØ8 ØD                 DEC     C
2BØ9 2ØØ8               JR      NZ,FISAM1B      ;Go if not load record
2BØB CDØB2D             CALL    $GET            ;  else read 1st two
2BØE Ø5                 DEC     B               ;  bytes & then fall thru
2BØF CDØB2D             CALL    $GET            ;  in case len=Ø1 or Ø2
2B12 Ø5                 DEC     B
2B13 78         FISAM1B LD      A,B
2B14 C1                 POP     BC
2B15 47                 LD      B,A
2B16 CDØB2D     FISAM2  CALL    $GET            ;Through the block
2B19 1ØFB               DJNZ    FISAM2
2B1B 18CF               JR      FISAM1          ;Go back for more
                ;
                ;       Found a map field
                ;
2B1D CDØB2D     FISAM3  CALL    $GET            ;Get field length
2B2Ø 47                 LD      B,A             ;Set counter
2B21 CDØB2D             CALL    $GET            ;Get overlay #
2B24 Ø5                 DEC     B               ;Reduce count
2B25 B9                 CP      C               ;Is this the one?
2B26 2ØEE               JR      NZ,FISAM2       ;Loop to next field
2B28 CDØB2D             CALL    $GET            ;Get lo-order traadr
2B2B CDØB2D             CALL    $GET            ;Get hi-order transfer
2B2E CDØB2D             CALL    $GET            ;Get lo-order NRN
2B31 4F                 LD      C,A             ;Save in C
2B32 CDØB2D             CALL    $GET            ;Get hi-order NRN
2B35 47                 LD      B,A             ;Save in B
2B36 CDØB2D             CALL    $GET            ;Get byte offset
2B39 C9                 RET
                ;
                ;       Routine to repair the library map
                ;
2B3A 117F2D     RPRMAP  LD      DE,PGMDCB       ;Rewind the file
```

```
2B3D 010000     07090          LD      BC,0
2B40 CDF52C     07100          CALL    $POSN
2B43 210048     07110          LD      HL,PGMDATA      ;Pt to buffer area
2B46 CD0B2D     07120 RPRMAP1  CALL    $GET            ;Read the map into buf
2B49 FE0A       07130          CP      0AH             ;End of table?
2B4B 2821       07140          JR      Z,RPRMAP3
2B4D 77         07150          LD      (HL),A          ;Save type code
2B4E CD0B2D     07160          CALL    $GET            ;Get length
2B51 47         07170          LD      B,A             ;Set counter
2B52 7E         07180          LD      A,(HL)          ;Reget the TYPE
2B53 23         07190          INC     HL              ;Bump where to stuf len
2B54 3D         07200          DEC     A               ;Is this a load record?
2B55 70         07210          LD      (HL),B          ;Put length in too
2B56 23         07220          INC     HL
2B57 200C       07230          JR      NZ,RPRMAP2      ;Go if other type
2B59 CD0B2D     07240          CALL    $GET            ;  else get two extra
2B5C 05         07250          DEC     B               ;  & adjust length in
2B5D 77         07260          LD      (HL),A          ;  case len = 01 or 02
2B5E 23         07270          INC     HL
2B5F CD0B2D     07280          CALL    $GET
2B62 05         07290          DEC     B
2B63 23         07300          INC     HL
2B64 77         07310          LD      (HL),A
2B65 CD0B2D     07320 RPRMAP2  CALL    $GET            ;Save member # & rest of
2B68 77         07330          LD      (HL),A          ;  data entries
2B69 23         07340          INC     HL
2B6A 10F9       07350          DJNZ    RPRMAP2
2B6C 18D8       07360          JR      RPRMAP1
                07370 ;
                07380 ;        Found end of table
                07390 ;
2B6E 77         07400 RPRMAP3  LD      (HL),A          ;Show map end
2B6F 210048     07410          LD      HL,PGMDATA      ;Pt to beginning
2B72 7E         07420 RPRMAP4  LD      A,(HL)          ;P/u type code
2B73 23         07430          INC     HL
2B74 46         07440          LD      B,(HL)          ;P/u length
2B75 23         07450          INC     HL
2B76 FE08       07460          CP      8               ;Map is type 8
2B78 2811       07470          JR      Z,RPRMAP6
2B7A FE0A       07480          CP      0AH             ;End of map?
2B7C CA3D2D     07490          JP      Z,NOVRLY        ;Should not have gotten
2B7F 3D         07500          DEC     A
2B80 2004       07510          JR      NZ,RPRMAP5
2B82 23         07520          INC     HL              ;You should know what
2B83 05         07530          DEC     B               ;  this is for by now
2B84 23         07540          INC     HL
2B85 05         07550          DEC     B
2B86 23         07560 RPRMAP5  INC     HL              ;Bypass this field
2B87 10FD       07570          DJNZ    RPRMAP5
2B89 18E7       07580          JR      RPRMAP4
                07590 ;
                07600 ;        Found a type 8, check if ISAM # matches
                07610 ;
2B8B 7E         07620 RPRMAP6  LD      A,(HL)          ;P/u member #
2B8C 23         07630          INC     HL
2B8D 05         07640          DEC     B               ;Count down
2B8E FE00       07650 OVRLY    CP      0               ;Compare to patched one
2B90 20F4       07660          JR      NZ,RPRMAP5      ;Keep reading until found
2B92 23         07670          INC     HL              ;Bypass transfer address
2B93 23         07680          INC     HL
2B94 5E         07690          LD      E,(HL)          ;P/u the position lo
```

```
2B95 23         Ø77ØØ          INC     HL
2B96 56         Ø771Ø          LD      D,(HL)          ;  & the pos hi
2B97 23         Ø772Ø          INC     HL
2B98 4E         Ø773Ø          LD      C,(HL)          ;  & the byte offset
2B99 78         Ø774Ø          LD      A,B             ;Calc ptr to next field
2B9A D6Ø4       Ø775Ø          SUB     4
2B9C 47         Ø776Ø          LD      B,A
2B9D 23         Ø777Ø          INC     HL
2B9E 1ØFD       Ø778Ø          DJNZ    $-1             ;Loop to next field
2BAØ 7E         Ø779Ø RPRMAP7  LD      A,(HL)          ;End of table?
2BA1 FEØA       Ø78ØØ          CP      ØAH             ;If end, write the
2BA3 2836       Ø781Ø          JR      Z,RWRMAP        ;  map back to disk
2BA5 23         Ø782Ø          INC     HL              ;Pt to field length
2BA6 46         Ø783Ø          LD      B,(HL)
2BA7 23         Ø784Ø          INC     HL              ;Pt to member #
2BA8 23         Ø785Ø          INC     HL              ;Transfer Low
2BA9 23         Ø786Ø          INC     HL              ;Transfer High
2BAA 23         Ø787Ø          INC     HL              ;NRN Low
2BAB 78         Ø788Ø          LD      A,B             ;Adjust count for
2BAC D6Ø4       Ø789Ø          SUB     4               ;  4 INC HLs
2BAE 47         Ø79ØØ          LD      B,A
2BAF 7E         Ø791Ø          LD      A,(HL)          ;If position is the same
2BBØ 23         Ø792Ø          INC     HL              ;  as that of patched
2BB1 BB         Ø793Ø          CP      E               ;  one, its posn has not
2BB2 2ØØF       Ø794Ø          JR      NZ,RPRMAP8      ;  changed, so don't
2BB4 7E         Ø795Ø          LD      A,(HL)          ;  change it
2BB5 23         Ø796Ø          INC     HL
2BB6 Ø5         Ø797Ø          DEC     B
2BB7 BA         Ø798Ø          CP      D               ;Cp the hi order
2BB8 2ØØB       Ø799Ø          JR      NZ,RPRMAP9
2BBA 7E         Ø8ØØØ          LD      A,(HL)
2BBB B9         Ø8Ø1Ø          CP      C               ;  and the offset
2BBC 2ØØ7       Ø8Ø2Ø          JR      NZ,RPRMAP9
2BBE 23         Ø8Ø3Ø LPFLD    INC     HL
2BBF 1ØFD       Ø8Ø4Ø          DJNZ    $-1             ;Loop to end of field
2BC1 18DD       Ø8Ø5Ø          JR      RPRMAP7
                Ø8Ø6Ø ;
                Ø8Ø7Ø ;        Add the patch length to each position vector
                Ø8Ø8Ø ;
2BC3 23         Ø8Ø9Ø RPRMAP8  INC     HL              ;Bump to offset byte
2BC4 Ø5         Ø81ØØ          DEC     B
2BC5 11ØØØØ     Ø811Ø RPRMAP9  LD      DE,Ø            ;P/u patch length
2BC8 7E         Ø812Ø          LD      A,(HL)          ;P/u offset & add
2BC9 83         Ø813Ø          ADD     A,E             ;Lo-order patch length
2BCA 77         Ø814Ø          LD      (HL),A
2BCB 2B         Ø815Ø          DEC     HL              ;Pt to NRN
2BCC 2B         Ø816Ø          DEC     HL
2BCD 7E         Ø817Ø          LD      A,(HL)          ;P/u NRN lo-order
2BCE 8A         Ø818Ø          ADC     A,D             ;Add to it
2BCF 77         Ø819Ø          LD      (HL),A
2BDØ 23         Ø82ØØ          INC     HL              ;Pt to pos hi order
2BD1 7E         Ø821Ø          LD      A,(HL)          ;P/u the hi
2BD2 CEØØ       Ø822Ø          ADC     A,Ø             ;Add in any carry
2BD4 77         Ø823Ø          LD      (HL),A
2BD5 23         Ø824Ø          INC     HL              ;Pt to next map field
2BD6 11ØØØØ     Ø825Ø          LD      DE,Ø
2BD9 18E3       Ø826Ø          JR      LPFLD           ;Loop
                Ø827Ø ;
                Ø828Ø ;        Routine to re-write the library map table
                Ø829Ø ;
2BDB 117F2D     Ø83ØØ RWRMAP   LD      DE,PGMDCB       ;Rewind the program file
```

```
2BDE 010000                LD      BC,0
2BE1 CDF52C                CALL    $POSN
2BE4 210048                LD      HL,PGMDATA      ;Pt to mapbuf start
2BE7 7E           RWRMAP1  LD      A,(HL)          ;Ret when we get to
2BE8 FE0A                  CP      0AH             ;  the map end type
2BEA C8                    RET     Z
2BEB 4F                    LD      C,A             ;Save the type
2BEC 23                    INC     HL
2BED CD112D                CALL    $PUT            ;Put the type
2BF0 7E                    LD      A,(HL)          ;P/u length
2BF1 23                    INC     HL
2BF2 47                    LD      B,A             ;Save as counter
2BF3 CD112D                CALL    $PUT            ;Put out the length
2BF6 0D                    DEC     C               ;Again, by now...
2BF7 200C                  JR      NZ,RWRMAP2
2BF9 7E                    LD      A,(HL)
2BFA 23                    INC     HL
2BFB CD112D                CALL    $PUT
2BFE 05                    DEC     B
2BFF 7E                    LD      A,(HL)
2C00 23                    INC     HL
2C01 CD112D                CALL    $PUT
2C04 05                    DEC     B
2C05 7E           RWRMAP2  LD      A,(HL)          ;Put block of code
2C06 23                    INC     HL
2C07 CD112D                CALL    $PUT
2C0A 10F9                  DJNZ    RWRMAP2
2C0C 18D9                  JR      RWRMAP1         ;Loop for more
                  ;
                  ;        This routine enters at PASS1. It does the first pass
                  ;         thru the fix data, and checks for parms as well
                  ;         as checking the Drr,bb and Frr,bb matches.
                  ;
2C0E 320627       SPASS2   LD      (PASS2),A       ;Flag pass 2
2C11 C3F726                JP      DOFIX           ;Start over
                  ;
2C14 FE2E         PASS1    CP      '.'             ;Comment line?
2C16 2825                  JR      Z,OK
2C18 FE03                  CP      ETX             ;End of fix data?
2C1A 28F2                  JR      Z,SPASS2        ;End of pass1
2C1C CBAF                  RES     5,A             ;Make Upper case
2C1E FE44                  CP      'D'             ;D line patch?
2C20 281E                  JR      Z,FCHK
2C22 FE52                  CP      'R'             ;Remove parm?
2C24 CA3A28                JP      Z,REMOVE
2C27 FE4F                  CP      'O'             ;Special O parm?
2C29 CAD128                JP      Z,OVERB
2C2C FE46                  CP      'F'             ;Find line data?
2C2E 280D                  JR      Z,OK
2C30 FE59                  CP      'Y'             ;Yank parm?
2C32 2809                  JR      Z,OK
2C34 FE4C                  CP      'L'             ;Library ISAM number?
2C36 2805                  JR      Z,OK
2C38 FE58                  CP      'X'             ;X line patch?
2C3A C2612D                JP      NZ,PCHERR       ;If not one of these, abort
2C3D C3C527       OK       JP      COMMENT
                  ;
                  ;        Check the Drr,bb (if Remove) or Frr,bb line
                  ;
2C40 3EFF         FCHK     LD      A,0FFH          ;If O parm = OFF, then
2C41              OPARM    EQU     $-1             ;  don't do the check
```

```
2C42 B7                          OR      A
2C43 CAC527                      JP      Z,COMMENT       ;Skip check if O=OFF
2C46 3E00                        LD      A,$-$           ;Remove parm used?
2C47                     RPARM   EQU     $-1
2C48 B7                          OR      A
2C49 C2862C                      JP      NZ,YANKD        ;Reverse D & F lines if so
2C4C 22772D                      LD      (DL),HL         ;Save D pointer
2C4F CDA02C                      CALL    SKPLN           ;Move to F line
2C52 CD582C                      CALL    DOCHK           ;Cp F line bytes w/file
2C55 C3FA26                      JP      DOFIX1
                         ;
                         ;
                         ;       Checks Drr,bb and Frr,bb addresses for a match
                         ;       Checks Frr,bb against program file if patching, or
                         ;        Drr,bb if removing
                         ;
2C58 226B2D              DOCHK   LD      (SETMSG+1),HL   ;Set line error msg
2C5B E5                          PUSH    HL              ;Save posn
2C5C ED5B772D                    LD      DE,(DL)         ;Get D or F line
2C60 0603                        LD      B,3             ;Init check count
2C62 23                  CP3     INC     HL
2C63 13                          INC     DE
2C64 1A                          LD      A,(DE)
2C65 BE                          CP      (HL)
2C66 C2C62C                      JP      NZ,FERROR       ;'FIND' error
2C69 10F7                        DJNZ    CP3             ;Check first 3 bytes
                         ;
2C6B 0603                        LD      B,3             ;Assume was 2 digit rec #
2C6D 3E2C                        LD      A,','           ;Comma?
2C6F BE                          CP      (HL)
2C70 2802                        JR      Z,CP5           ;Yes, continue
2C72 0605                        LD      B,5             ;Adjust, assume 4 dig rec #
2C74 23                  CP5     INC     HL              ;Check rest of 'rr,bb' string
2C75 13                          INC     DE
2C76 1A                          LD      A,(DE)
2C77 BE                          CP      (HL)
2C78 C2C62C                      JP      NZ,FERROR       ;'FIND' error
2C7B 10F7                        DJNZ    CP5
                         ;
2C7D E3                          EX      (SP),HL         ;Pointer to '=' in fix line
2C7E CDE327                      CALL    DPOSN           ;Posn file
2C81 E1                          POP     HL
2C82 CD0A28                      CALL    DLINE           ;Check line for match
2C85 C9                          RET
                         ;
                         ;       Remove used. Check Drr,bb lines instead of Frr,bb lines
                         ;
2C86 E5                  YANKD   PUSH    HL              ;Save D line pointer
2C87 CDA02C                      CALL    SKPLN           ;Move to F line
2C8A 22772D                      LD      (DL),HL         ;Save pointer
2C8D E1                          POP     HL              ;=>D line
2C8E E5                          PUSH    HL              ;Save D line again
2C8F CD582C                      CALL    DOCHK           ;Test D line
2C92 E1                          POP     HL              ;=>'D'
2C93 362E                        LD      (HL),'.'        ;Make comment for pass2
2C95 2A772D                      LD      HL,(DL)
2C98 3644                        LD      (HL),'D'        ;Make 'F' line into D line
2C9A CDA02C                      CALL    SKPLN           ;=>next line
2C9D C3FA26                      JP      DOFIX1
                         ;
                         ;       Skip past the current line, posn to start of next
```

```
                ;
2CAØ CDA92C     SKPLN   CALL    SKPLN1          ;Move past current line
2CA3 7E                 LD      A,(HL)          ;Check 1st char next line
2CA4 FE2E               CP      '.'             ;Is it comment?
2CA6 28F8               JR      Z,SKPLN         ;Then skip it too
2CA8 C9                 RET
                ;
2CA9 7E         SKPLN1  LD      A,(HL)          ;P/u line char
2CAA 23                 INC     HL
2CAB FEØD               CP      CR              ;Physical EOL?
2CAD C8                 RET     Z
2CAE FE3B               CP      ';'             ;Logical EOL?
2CBØ C8                 RET     Z
2CB1 18F6               JR      SKPLN1          ;Loop until EOL
                ;
                ;       Get the next char, convert to UC
                ;
2CB3 7E         GETNXT  LD      A,(HL)          ;P/u the char
2CB4 23                 INC     HL              ;Bump the buffer ptr
2CB5 CBAF               RES     5,A             ;Convert to upper
2CB7 C9                 RET
                ;
                ;       Either write a char or check for a match
                ;
2CB8 4F         PUTORCHK        LD      C,A     ;Char in question
2CB9 3AØ627             LD      A,(PASS2)       ;Write pass?
2CBC B7                 OR      A
2CBD 79                 LD      A,C             ;Char back in A
2CBE C2112D             JP      NZ,$PUT         ;Writing patch..
2CC1 CDØB2D             CALL    $GET            ;Get next char fm file
2CC4 B9                 CP      C               ;Match w/patch?
2CC5 C8                 RET     Z               ;OK if match
2CC6 21113Ø     FERROR  LD      HL,LOCERR$      ;Init "Find mismatch
2CC9 C3642D             JP      ERRDSP          ;Dsply and quit
                ;
                ;       Count patch lines
                ;
2CCC E5         CNTLIN  PUSH    HL
2CCD 2A742D             LD      HL,(LINCNT)     ;Get current count,
2CDØ 23                 INC     HL              ; += 1
2CD1 22742D             LD      (LINCNT),HL     ;  and put it back
2CD4 E1                 POP     HL
2CD5 C9                 RET
                ;
                ;       After an error, show file not closed if needed
                ;
2CD6 3A762D     FLOPN   LD      A,(WRFLAG)      ;Did we modify file?
2CD9 B7                 OR      A
2CDA 2ØØ7               JR      NZ,MESS         ;Yes, don't close it
2CDC                    @@CLOSE                 ;No changes
2CDC 3E3C               LD      A,6Ø
2CDE EF                 RST     4Ø
2CDF C2322D             JP      NZ,IOERR
2CE2 C9                 RET
2CE3 21243Ø     MESS    LD      HL,WARN1$       ;File is modified but
2CE6 CD1F2D             CALL    $DSPLY          ;PATCH did not complete
2CE9 214B3Ø             LD      HL,WARN2$       ; "oops...
2CEC C31F2D             JP      $DSPLY          ;Then return to caller
                ;
2CEF            $OPEN   @@OPEN
2CEF 3E3B               LD      A,59
```

```
2CF1 EF                      RST     4Ø
2CF2 2Ø3E                    JR      NZ,IOERR
2CF4 C9                      RET
2CF5             $POSN       @@POSN
2CF5 3E42                    LD      A,66
2CF7 EF                      RST     4Ø
2CF8 2Ø38                    JR      NZ,IOERR
2CFA C9                      RET
2CFB             $BKSP       @@BKSP
2CFB 3E3D                    LD      A,61
2CFD EF                      RST     4Ø
2CFE 2Ø32                    JR      NZ,IOERR
2DØØ C9                      RET
2DØ1             $RWRIT      @@RWRIT
2DØ1 3E46                    LD      A,7Ø
2DØ3 EF                      RST     4Ø
2DØ4 2Ø2C                    JR      NZ,IOERR
2DØ6 C9                      RET
2DØ7             $GET1       @@GET                   ;Use this one if prog might get EOF
2DØ7 3EØ3                    LD      A,3
2DØ9 EF                      RST     4Ø
2DØA C9                      RET
2DØB             $GET        @@GET                   ;This one if EOF is also error
2DØB 3EØ3                    LD      A,3
2DØD EF                      RST     4Ø
2DØE 2Ø22                    JR      NZ,IOERR
2D1Ø C9                      RET
2D11 C5          $PUT        PUSH    BC
2D12 4F                      LD      C,A
2D13 3EFF                    LD      A,ØFFH          ;Flag..
2D15 32762D                  LD      (WRFLAG),A      ;That file is modified
2D18                         @@PUT
2D18 3EØ4                    LD      A,4
2D1A EF                      RST     4Ø
2D1B C1                      POP     BC
2D1C 2Ø14                    JR      NZ,IOERR
2D1E C9                      RET
2D1F             $DSPLY      @@DSPLY
                             IFEQ    ØØH,1
                             LD      HL,
                             ENDIF
2D1F 3EØA                    LD      A,1Ø
2D21 EF                      RST     4Ø
2D22 2ØØE                    JR      NZ,IOERR
2D24 C9                      RET
2D25             $READ       @@READ
2D25 3E43                    LD      A,67
2D27 EF                      RST     4Ø
2D28 2ØØ8                    JR      NZ,IOERR
2D2A C9                      RET
2D2B C5          $DSP        PUSH    BC
2D2C 4F                      LD      C,A
2D2D                         @@DSP
2D2D 3EØ2                    LD      A,2
2D2F EF                      RST     4Ø
2D3Ø C1                      POP     BC
2D31 C8                      RET     Z               ;If OK else fall error
                 ;
                 ;           Error handling
                 ;
2D32 6F          IOERR       LD      L,A             ;HL also gets error #
```

```
2D33 2600      10500          LD      H,0
2D35 F6C0      10510          OR      0C0H           ;Abbrev, return
2D37 4F        10520          LD      C,A
2D38           10530          @@ERROR                ;Display the error
2D38 3E1A      00060          LD      A,26
2D3A EF        00061          RST     40
2D3B 181D      10540          JR      QUIT1
               10550 ;
               10560 ;        Internal error routine
               10570 ;
2D3D 211B2F    10580 NOVRLY   LD      HL,NOVRLY$     ;"Library not found
2D40 DD        10590          DB      0DDH
2D41 21352F    10600 LIBERR   LD      HL,LIBERR$     ;"Invalid library
2D44 DD        10610          DB      0DDH
2D45 21652F    10620 FILERR   LD      HL,FILERR$     ;"Not load file format
2D48 DD        10630          DB      0DDH
2D49 215F2E    10640 PRMERR   LD      HL,PRMERR$     ;"Parm error
2D4C DD        10650          DB      0DDH
2D4D 21CE2F    10660 TOOBIG   LD      HL,TOOBIG$     ;"Fix file too big
2D50 DD        10670          DB      0DDH
2D51 21442E    10680 PGMREQ   LD      HL,PGMREQ$     ;"Patch what file?
2D54           10690 ERREXIT @@LOGOT                 ;Display the error
               00062          IFEQ    00H,1
               00063          LD      HL,
               00064          ENDIF
2D54 3E0C      00065          LD      A,12
2D56 EF        00066          RST     40
2D57 21FFFF    10700          LD      HL,-1          ;Set abort code
2D5A C3AE27    10710 QUIT1    JP      $QUIT
               10720 ;
2D5D 217C2F    10730 NONHEX   LD      HL,NONHEX$     ;"Non hex digit
2D60 DD        10740          DB      0DDH
2D61 214C2F    10750 PCHERR   LD      HL,PCHERR$     ;"Patch format error
2D64 E5        10760 ERRDSP   PUSH    HL
2D65 3E0D      10770          LD      A,CR           ;Move the cursor down
2D67 CD2B2D    10780          CALL    $DSP
2D6A 210000    10790 SETMSG   LD      HL,0
2D6D           10800          @@LOGOT
               00067          IFEQ    00H,1
               00068          LD      HL,
               00069          ENDIF
2D6D 3E0C      00070          LD      A,12
2D6F EF        00071          RST     40
2D70 E1        10810          POP     HL
2D71 18E1      10820          JR      ERREXIT
               10830 ;
2D73 00        10840 YNKFLG   DB      0              ;Was function YANK?
2D74 0000      10850 LINCNT   DW      0              ;Count lines installed
2D76 00        10860 WRFLAG   DB      0              ;Did pgm write to file?
2D77 0000      10870 DL       DW      0              ;Save pointer to line
2D79 43        10880 CMDEXT   DB      'CMD'
     4D 44
2D7C 46        10890 FIXEXT   DB      'FIX'
     49 58
2D7F 00        10900 PGMDCB   DB      0
0020           10910          DS      32
0020           10920 FIXDCB   DS      32
2DC0 50        10930 HELLO$   DB      'PATCH'
     41 54 43 48
2DC5           10940 *GET     CLIENT:3
               10950 ;CLIENTS/ASM - File to establish sign-on headers
```

```
                10960 ;
2DC5 20         10970          DB      ' - 6.2.0 - Copyright 1982/83/84 by Logical'
     2D 20 36 2E 32 2E 30 20
     2D 20 43 6F 70 79 72 69
     67 68 74 20 31 39 38 32
     2F 38 33 2F 38 34 20 62
     79 20 4C 6F 67 69 63 61
     6C
2DEF 20         10980          DB      ' Systems, Inc.      ',10
     53 79 73 74 65 6D 73 2C
     20 49 6E 63 2E 20 20 20
     20 20 20 0A
                10990 ;
2E04 41         11000          DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 20 52 69 67 68 74
     73 20 52 65 73 65 72 76
     65 64 2E 20 4C 69 63 65
     6E 73 65 64 20 31 39 38
     32 2F 38 33 2F 38 34
2E2C 20         11010          DB      ' to xxxxxxxxxxxxxxxxxx',10,13
     74 6F 20 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 0A 0D
2E44 50         11020 PGMREQ$ DB       'PROGRAM file name required',CR
     52 4F 47 52 41 4D 20 66
     69 6C 65 20 6E 61 6D 65
     20 72 65 71 75 69 72 65
     64 0D
2E5F 50         11030 PRMERR$ DB       'Parameter error',CR
     61 72 61 6D 65 74 65 72
     20 65 72 72 6F 72 0D
2E6F 1D         11040 POSLD$  DB       29,'Positioning load file',30,32,3
     50 6F 73 69 74 69 6F 6E
     69 6E 67 20 6C 6F 61 64
     20 66 69 6C 65 1E 20 03
2E88 1D         11050 RDGINP$ DB       29,'Reading input',30,32,3
     52 65 61 64 69 6E 67 20
     69 6E 70 75 74 1E 20 03
2E99 1D         11060 GENPCH$ DB       29,'Generating patch',30,32,3
     47 65 6E 65 72 61 74 69
     6E 67 20 70 61 74 63 68
     1E 20 03
2EAD 1D         11070 INSPCH$ DB       29,'Installing patch',30,32,3
     49 6E 73 74 61 6C 6C 69
     6E 67 20 70 61 74 63 68
     1E 20 03
2EC1 1D         11080 BLDMAP$ DB       29,'Re-building library map',30,32,3
     52 65 2D 62 75 69 6C 64
     69 6E 67 20 6C 69 62 72
     61 72 79 20 6D 61 70 1E
     20 03
2EDC 1D         11090 YNKPCH$ DB       29,'Yanking patch from file',30,32,3
     59 61 6E 6B 69 6E 67 20
     70 61 74 63 68 20 66 72
     6F 6D 20 66 69 6C 65 1E
     20 03
2EF7 0A         11100 NOYANK$ DB       LF,'Can''t yank, '
     43 61 6E 27 74 20 79 61
     6E 6B 2C 20
2F04 70         11110          DB      'patch not in load file',CR
     61 74 63 68 20 6E 6F 74
     2Ø 69 6E 2Ø 6C 6F 61 64
     2Ø 66 69 6C 65 ØD
2F1B 4C          1112Ø NOVRLY$ DB       'Library overlay not found',CR
     69 62 72 61 72 79 2Ø 6F
     76 65 72 6C 61 79 2Ø 6E
     6F 74 2Ø 66 6F 75 6E 64
     ØD
2F35 49          1113Ø LIBERR$ DB       'Invalid library format',CR
     6E 76 61 6C 69 64 2Ø 6C
     69 62 72 61 72 79 2Ø 66
     6F 72 6D 61 74 ØD
2F4C 5Ø          1114Ø PCHERR$ DB       'Patch input format error',CR
     61 74 63 68 2Ø 69 6E 7Ø
     75 74 2Ø 66 6F 72 6D 61
     74 2Ø 65 72 72 6F 72 ØD
2F65 4C          1115Ø FILERR$ DB       'Load file format error',CR
     6F 61 64 2Ø 66 69 6C 65
     2Ø 66 6F 72 6D 61 74 2Ø
     65 72 72 6F 72 ØD
2F7C 4E          1116Ø NONHEX$ DB       'Non-hex digit encountered',CR
     6F 6E 2D 68 65 78 2Ø 64
     69 67 69 74 2Ø 65 6E 63
     6F 75 6E 74 65 72 65 64
     ØD
2F96 ØA          1117Ø SUCCES$ DB       LF,'Patch function completed.',CR
     5Ø 61 74 63 68 2Ø 66 75
     6E 63 74 69 6F 6E 2Ø 63
     6F 6D 7Ø 6C 65 74 65 64
     2E ØD
2FB1 2Ø          1118Ø LINMSG$ DB       '   No patch line'
     2Ø 2Ø 4E 6F 2Ø 7Ø 61 74
     63 68 2Ø 6C 69 6E 65
2FC1 73          1119Ø PLURAL  DB       's installed.',CR
     2Ø 69 6E 73 74 61 6C 6C
     65 64 2E ØD
2FCE 46          112ØØ TOOBIG$ DB       'Fix file too big - partition it',CR
     69 78 2Ø 66 69 6C 65 2Ø
     74 6F 6F 2Ø 62 69 67 2Ø
     2D 2Ø 7Ø 61 72 74 69 74
     69 6F 6E 2Ø 69 74 ØD
2FEE 5Ø          1121Ø YANKMSG DB       'Patch successfully yanked',CR
     61 74 63 68 2Ø 73 75 63
     63 65 73 73 66 75 6C 6C
     79 2Ø 79 61 6E 6B 65 64
     ØD
3ØØ8 Ø3          1122Ø NAMLEN$ DB       3               ;Length of fix file name
3ØØ9 43          1123Ø NAMFIX$ DB       'CLP     '      ;Fix file name
     4C 5Ø 2Ø 2Ø 2Ø 2Ø 2Ø
3Ø11 46          1124Ø LOCERR$ DB       'FIND line mismatch',CR
     49 4E 44 2Ø 6C 69 6E 65
     2Ø 6D 69 73 6D 61 74 63
     68 ØD
3Ø24 57          1125Ø WARN1$  DB       'WARNING - File '
     41 52 4E 49 4E 47 2Ø 2D
     2Ø 46 69 6C 65 2Ø
3Ø33 2Ø          1126Ø FNM$    DB       '                        '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
3Ø4B 2Ø          1127Ø WARN2$  DB       ' Not Closed',CR
     4E 6F 74 2Ø 43 6C 6F 73
     65 64 ØD
              1128Ø ;
3Ø57 8Ø       1129Ø PTBL$   DB      8ØH
3Ø58 56       113ØØ         DB      FLAG!ABB!6
3Ø59 52       1131Ø         DB      'REMOVE',Ø
     45 4D 4F 56 45 ØØ
3Ø6Ø 8D26     1132Ø         DW      RPARM1
3Ø62 41       1133Ø         DB      FLAG!1
3Ø63 4F       1134Ø         DB      'O',Ø
     ØØ
3Ø65 9426     1135Ø         DW      OPARM1
3Ø67 ØØ       1136Ø         NOP
              1137Ø ;
31ØØ          1138Ø         ORG     $<-8+1<+8
Ø1ØØ          1139Ø FIXBUF  DS      256             ;I/O buffer for /FIX
Ø1ØØ          114ØØ LIBBUF  DS      256             ;I/O buffer for ISAM
Ø1ØØ          1141Ø PGMBUF  DS      256             ;I/O buffer for PGM
14ØØ          1142Ø FIXDATA DS      14ØØH           ;5k alloted for fix data
48ØØ          1143Ø PGMDATA EQU     $               ;Takes the rest of core
              Ø51ØØ ;
26ØØ          Ø511Ø         END     BEGIN
```

```
$BKSP          2CFB $DSP          2D2B $DSPLY         2D1F
$GET           2D0B $GET1         2D07 $OPEN          2CEF
$POSN          2CF5 $PUT          2D11 $QUIT          27AE
$READ          2D25 $RWRIT        2D01 @@1            0000
@@2            0000 @@3           0000 @@4            0000
@MOD2          0000 @MOD4         FFFF ABB            0010
BEGIN          2600 BEGINA        2609 BLDMAP$        2EC1
CKLIN          2674 CKLIN1        26A0 CKLIN2         26B1
CKLIN3         26B5 CMDEXT        2D79 CNTLIN         2CCC
COMMENT        27C5 CP3           2C62 CP5            2C74
CR             000D CVTBIN        2AD7 DL             2D77
DLINE          280A DOCHK         2C58 DOFIX          26F7
DOFIX1         26FA DOXVB         29C4 DOXVB1         29EB
DOXVB2         29EE DOXVB3        29F3 DOXVB4         2A0A
DOXVB5         2A12 DOXVB6        2A29 DPOSN          27E3
DVERB          27D7 DVERB1        27F3 DVERB2         2810
DVERB3         2811 DVERB4        282C DVERB4A        282D
EOL1           27D4 ERRDSP        2D64 ERREXIT        2D54
ETX            0003 EXLOG         27A8 FCHK           2C40
FERROR         2CC6 FILERR        2D45 FILERR$        2F65
FISAM          2AE9 FISAM1        2AEC FISAM1A        2AF9
FISAM1B        2B13 FISAM2        2B16 FISAM3         2B1D
FIXBUF         3100 FIXDATA       3400 FIXDCB         2DA0
FIXEXT         2D7C FLAG          0040 FLOPN          2CD6
FNM$           3033 FXNAM         2658 FXNAM1         266A
FXNAM2         266F GENPCH$       2E99 GETNXT         2CB3
HELLO$         2DC0 INSPCH$       2EAD IOERR          2D32
LF             000A LIBBUF        3200 LIBERR         2D41
LIBERR$        2F35 LINCNT        2D74 LINMSG$        2FB1
LOCERR$        3011 LPFLD         2BBE LVERB          2900
MESS           2CE3 NAMFIX$       3009 NAMLEN$        3008
NOCHG          27A5 NONHEX        2D5D NONHEX$        2F7C
NOVRLY         2D3D NOVRLY$       2F1B NOYANK$        2EF7
NTONE          279F OISOFF        28F8 OISON          28F7
OK             2C3D OPARM         2C41 OPARM1         2694
OVERB          28D1 OVRLY         2B8E PASS1          2C14
PASS2          2706 PCHDUN        276D PCHERR         2D61
PCHERR$        2F4C PGMBUF        3300 PGMDATA        4800
PGMDCB         2D7F PGMREQ        2D51 PGMREQ$        2E44
PLURAL         2FC1 POSFIL        294D POSFIL1        2957
POSFIL2        297B POSLD$        2E6F PRMERR         2D49
PRMERR$        2E5F PRSFIX        2A92 PRSFX1         2A96
PRSFX2         2AC7 PRSFX3        2AD4 PTBL$          3057
PUTORCHK       2CB8 QUIT1         2D5A RDFIX          26BE
RDFIX1         26DB RDFIX2        26EF RDFIX3         26F4
RDGINP$        2E88 REMOVE        283A RIPPL1         2A5D
RIPPL2         2A7A RIPPL3        2A86 RIPPLE         2A2F
RPARM          2C47 RPARM1        268D RPRMAP         2B3A
RPRMAP1        2B46 RPRMAP2       2B65 RPRMAP3        2B6E
RPRMAP4        2B72 RPRMAP5       2B86 RPRMAP6        2B8B
RPRMAP7        2BA0 RPRMAP8       2BC3 RPRMAP9        2BC5
RWRMAP         2BDB RWRMAP1       2BE7 RWRMAP2        2C05
SETMSG         2D6A SKPLN         2CA0 SKPLN1         2CA9
SPASS2         2C0E STACK         27BF STRFLG         2AAA
STUFNM         2982 STUFNM1       2999 STUFNM2        299F
STUFNM3        29A0 STUFNM4       29B7 SUCCES$        2F96
TOOBIG         2D4D TOOBIG$       2FCE TYPCOD         2869
WARN1$         3024 WARN2$        304B WHATIS         28F4
WRFLAG         2D76 YANK          2842 YANK1          2858
YANK2          2876 YANK3         2879 YANK4          287D
```

```
YANK5          288A YANK6          2896 YANK7          28B6
YANK8          28BD YANK9          28C6 YANKD          2C86
YANKMSG        2FEE YNKFLG         2D73 YNKPCH$        2EDC
@@ABORT        AØ88 @@ADTSK        A11B @@BANK         A633
@@BKSP         A313 @@BREAK        A649 @@CHNIO        AØ73
@@CKBRKC       A697 @@CKDRV        A16F @@CKEOF        A328
@@CKTSK        A1Ø6 @@CLOSE        A2FE @@CLS          A681
@@CMNDI        AØB2 @@CMNDR        AØC7 @@CTL          9ED7
@@DATE         AØ49 @@DCSTAT       A1AE @@DEBUG        AØF1
@@DECHEX       A5B3 @@DIRRD        A52Ø @@DIRWR        A535
@@DIV16        A59E @@DIV8         A589 @@DODIR        A184
@@DSP          9E9B @@DSPLY        9F3B @@ERROR        AØDC
@@EXIT         AØ9D @@FEXT         A48D @@FLAGS        A61D
@@FNAME        A4A2 @@FSPEC        A478 @@GATRD        A5ØB
@@GATWR        A54A @@GET          9EAF @@GTDCB        A4CC
@@GTDCT        A4B7 @@GTMOD        A4E1 @@HDFMT        A256
@@HEX16        A5F2 @@HEX8         A5DD @@HEXDEC       A5C8
@@HIGH$        A6Ø7 @@INIT         A2D4 @@KBD          9F13
@@KEY          9E87 @@KEYIN        9F27 @@KLTSK        A15A
@@LOAD         A44E @@LOC          A33D @@LOF          A352
@@LOGER        9F72 @@LOGOT        9F87 @@MSG          9FBE
@@MUL16        A574 @@MUL8         A55F @@OPEN         A2E9
@@PARAM        AØ34 @@PAUSE        AØ1F @@PEOF         A367
@@POSN         A37C @@PRINT        9FD3 @@PRT          9EEB
@@PUT          9EC3 @@RAMDIR       A199 @@RDSEC        A22C
@@RDSSC        A4F6 @@READ         A391 @@REMOV        A2BF
@@RENAM        A2AA @@REW          A3A6 @@RMTSK        A13Ø
@@RPTSK        A145 @@RREAD        A3BB @@RSLCT        A217
@@RSTOR        A1D8 @@RUN          A463 @@RWRIT        A3DØ
@@SEEK         A2Ø2 @@SEEKSC       A3E5 @@SKIP         A3FA
@@SLCT         A1C3 @@STEPI        A1ED @@TIME         AØ5E
@@VDCTL        AØØA @@VER          A4ØF @@VRSEC        A241
@@WEOF         A424 @@WHERE        9EFF @@WRITE        A439
@@WRSEC        A26B @@WRSSC        A28Ø @@WRTRK        A295

26ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

REPAIR/CMD - Repair a disk directory cylinder

The Repair utility will write the directory cylinder with the proper data address mark, and update certain information in the GAT that is needed by LS-DOS. Its main use is to make Model I TRSDOS disks readable by LS-DOS/TRSDOS 6.

```
             ;REPAIR/ASM - Directory Track Repair Program
0000                 TITLE   <REPAIR - LS-DOS 6.2>
             ;
000A         LF      EQU     10
000D         CR      EQU     13
4296         BLNKMPW EQU     4296H
0040         FLAG    EQU     01000000B
0010         ABB     EQU     00010000B
             ;
0000         *GET    SVCMAC:3                ;SVC Macro equivalents
             ;SVCMAC/ASM - LS-DOS Version VI
             *LIST   OFF
             *LIST   ON
0000         *GET    COPYCOM:3               ;Copyright message
             ; COPYCOM - File for Copyright COMment block
             ;
0000                 COM     '<*(C) 1982,83,84 by LSI*>'
             ;
2600                 ORG     2600H
             BEGIN
2600                 @@CKBRKC                ;Check for break
2600 3E6A            LD      A,106
2602 EF              RST     40
2603 2804            JR      Z,BEGINA        ;Continue if not
2605 21FFFF          LD      HL,-1           ;  else abort
2608 C9              RET
             ;
             BEGINA
2609 ED731C26        LD      (STACK),SP      ;Save entry stack
260D E5              PUSH    HL              ;Save ptr to CMD buffer
260E 214A28          LD      HL,HELLO$       ;Display the signon
2611 CD9F27          CALL    $DSPLY
2614 E1              POP     HL
2615 CD2226          CALL    PGRM            ;Normal exit is via RET
             ;
             ;       Set exit condition..
             ;
2618 210000  $EXIT   LD      HL,0            ;Init for no error
261B 310000  QUIT$   LD      SP,$-$          ;P/u original stack
261C         STACK   EQU     $-2
261E                 @@CKBRKC                ;Clear break before exit
261E 3E6A            LD      A,106
2620 EF              RST     40
2621 C9              RET
             ;
2622 7E      PGRM    LD      A,(HL)          ;Ck for drive entered
2623 FE3A            CP      ':'             ;Colon indicator?
2625 C2E127          JP      NZ,PRMERR       ;Quit if not
2628 23              INC     HL              ;Point to drive #
2629 7E              LD      A,(HL)          ;P/u drive
262A D630            SUB     '0'             ;Cvrt to binary
262C FE08            CP      8               ;Bigger than 7?
262E D24528          JP      NC,ILLEG        ;Quit if so
             ;
2631 B7              OR      A               ;Can't be drive 0
2632 CADD27          JP      Z,NOT0
2635 321C27          LD      (DRIVE),A       ;Stuff for later
2638 23              INC     HL              ;Bump past the drive
2639 4F              LD      C,A
263A                 @@GTDCT                 ;What's its DCT$
263A 3E51            LD      A,81
```

```
263C EF        ØØØØ6          RST     4Ø
               ØØ6ØØ ;
               ØØ61Ø ;        Get any parameters
               ØØ62Ø ;
263D 114E29    ØØ63Ø          LD      DE,PRMTBL$       ;Pt to parm table
264Ø           ØØ64Ø          @@PARAM
264Ø 3E11      ØØØØ7          LD      A,17
2642 EF        ØØØØ8          RST     4Ø
2643 C2E127    ØØ65Ø          JP      NZ,PRMERR        ;Exit on parm error
2646 3A5329    ØØ66Ø          LD      A,(MRSP)         ;MPW parameter entered?
2649 B7        ØØ67Ø          OR      A
264A C2Ø627    ØØ68Ø          JP      NZ,MPARM         ;Go if so
264D FDCBØ35E  ØØ69Ø          BIT     3,(IY+3)         ;Can't "repair" a hard drive
2651 C2D927    ØØ7ØØ          JP      NZ,NIXHARD       ;  except for MPW parm
2654 FDCBØ466  ØØ71Ø          BIT     4,(IY+4)         ;If not alien controller
2658 CC1128    ØØ72Ø          CALL    Z,CKDRV          ;  make sure disk present
265B 11ØØØØ    ØØ73Ø          LD      DE,Ø             ;Read BOOT to get dir cyl
265E CDB827    ØØ74Ø          CALL    RDSEC
2661 AF        ØØ75Ø          XOR     A
2662 32ØØ2A    ØØ76Ø          LD      (BUF1),A         ;Set 1st byte to zero
2665 3AØ22A    ØØ77Ø          LD      A,(BUF1+2)       ;P/u the dir cyl
2668 E67F      ØØ78Ø          AND     7FH              ;Strip bit 7
266A 32Ø22A    ØØ79Ø          LD      (BUF1+2),A       ;Put it back
266D F5        ØØ8ØØ          PUSH    AF               ;Save dir cyl
266E CDB227    ØØ81Ø          CALL    WRSEC            ;Rewrite the BOOT
2671 1C        ØØ82Ø          INC     E
2672 CDB827    ØØ83Ø          CALL    RDSEC            ;Get sect 1 also
2675 F1        ØØ84Ø          POP     AF
2676 32Ø22A    ØØ85Ø          LD      (BUF1+2),A       ;Update dir cyl
2679 F5        ØØ86Ø          PUSH    AF
267A CDB227    ØØ87Ø          CALL    WRSEC            ;Write back
267D F1        ØØ88Ø          POP     AF               ;Dir cyl again
               ØØ89Ø ;
267E 57        ØØ9ØØ          LD      D,A
267F 1EØØ      ØØ91Ø          LD      E,Ø
2681 FD77Ø9    ØØ92Ø          LD      (IY+9),A         ;Set as dir cyl
2684 CDB827    ØØ93Ø          CALL    RDSEC            ;Read the GAT
               ØØ94Ø ;
2687 FDCBØ4AE  ØØ95Ø          RES     5,(IY+4)         ;Show single sided
268B 2ECB      ØØ96Ø          LD      L,ØCBH           ;Pt to version # byte
268D 7E        ØØ97Ø          LD      A,(HL)           ;Pick it up
268E FE4Ø      ØØ98Ø          CP      4ØH              ;Earlier than a 4.Ø?
269Ø 38ØE      ØØ99Ø          JR      C,LC             ;Bypass 2 sided ck if so
2692 FE7Ø      Ø1ØØØ          CP      7ØH              ;"Later" than 6.x?
2694 3ØØA      Ø1Ø1Ø          JR      NC,LC            ;Again, no sides ck
2696 2ECD      Ø1Ø2Ø          LD      L,ØCDH           ;Point to CONFIG byte
2698 CB6E      Ø1Ø3Ø          BIT     5,(HL)           ;Check 2-sided
269A 28Ø4      Ø1Ø4Ø          JR      Z,LC             ;Go if not
269C FDCBØ4EE  Ø1Ø5Ø          SET     5,(IY+4)         ;  else update DCT
               Ø1Ø6Ø ;
26AØ 2EBF      Ø1Ø7Ø LC       LD      L,ØBFH           ;Pt to end of lockout
26A2 Ø66Ø      Ø1Ø8Ø          LD      B,96             ;Max cylinder count
26A4 7E        Ø1Ø9Ø ALIEN1   LD      A,(HL)           ;P/u a lockout byte
26A5 3C        Ø11ØØ          INC     A                ;Locked out?
26A6 2ØØ3      Ø111Ø          JR      NZ,ALIEN2        ;Exit when in use
26A8 2D        Ø112Ø          DEC     L                ;Backup by 1
26A9 1ØF9      Ø113Ø          DJNZ    ALIEN1
26AB 3EDD      Ø114Ø ALIEN2   LD      A,-35            ;What's in use?
26AD 8Ø        Ø115Ø          ADD     A,B              ;Convert to excess
26AE 2ECC      Ø116Ø          LD      L,ØCCH
26BØ 77        Ø117Ø          LD      (HL),A           ;Stuff into GAT
```

```
                 ;
                 ;       Construct config byte
                 ;
26B1 FD7E04              LD      A,(IY+4)          ;P/u # sides
26B4 E6A0                AND     80H!20H
26B6 47                  LD      B,A               ;Save tempy
26B7 FD7E03              LD      A,(IY+3)          ;P/u density
26BA E640                AND     40H
26BC B0                  OR      B                 ;Merge with previous
26BD 47                  LD      B,A
26BE FD7E08              LD      A,(IY+8)          ;P/u # grans/track
26C1 07                  RLCA
26C2 07                  RLCA                      ;  to bits 0-2
26C3 07                  RLCA
26C4 E607                AND     7                 ;Mask off the rest
26C6 B0                  OR      B                 ;Merge with previous
26C7 2C                  INC     L                 ;Pt to config byte in GAT
26C8 47                  LD      B,A               ;Save for a moment
26C9 7E                  LD      A,(HL)            ;P/u present config byte
26CA E680                AND     80H               ;Keep only bit 7
26CC B0                  OR      B                 ;Pick up the rest
26CD 77                  LD      (HL),A            ;  & stuff
26CE 2E00                LD      L,0
26D0 CDA527              CALL    WRSYS             ;Write the GAT
                 ;
                 ;       Operate on the HIT
                 ;
26D3 1C                  INC     E                 ;Bump sector ptr to 1
26D4 CDB827              CALL    RDSEC             ;Read the HIT
26D7 2C                  INC     L                 ;Pt to DIR/SYS dec
26D8 36C4                LD      (HL),0C4H         ;"correct" DEC code
26DA 2D                  DEC     L
26DB CDA527              CALL    WRSYS             ;Write out the HIT
26DE 0608                LD      B,8               ;Init for 8 sectors
26E0 1C          ALIEN3  INC     E                 ;Bump to next sector
26E1 CDB827              CALL    RDSEC             ;Get the sector
26E4 CD8F27              CALL    UNOPEN            ;Reset file open bit
26E7 7B                  LD      A,E               ;If DIR/SYS sector,
26E8 FE03                CP      3                 ;  then update count & it
26EA 200E                JR      NZ,ALIEN4
26EC E5                  PUSH    HL
26ED 219642              LD      HL,BLNKMPW        ;Set DIR/SYS password
26F0 22122A              LD      (BUF1+12H),HL     ;To blanks
26F3 3A142A              LD      A,(BUF1+20)       ;P/u ERN of DIR/SYS
26F6 D603                SUB     3                 ;Account for 1st 3 done
26F8 47                  LD      B,A               ;Update loop counter
26F9 E1                  POP     HL
26FA CDA527      ALIEN4  CALL    WRSYS             ;Write back the sector
26FD 10E1                DJNZ    ALIEN3
                 ;
26FF                     @@LOGOT ALCAO$            ;Advise complete - now readable
                         IFEQ    01H,1
26FF 21CF28              LD      HL,ALCAO$
                         ENDIF
2702 3E0C                LD      A,12
2704 EF                  RST     40
2705 C9                  RET                       ;Done
                 ;
                 ;       MPW parameter to change disk password on hard drive
                 ;
2706 110000      MPARM   LD      DE,0              ;P/u MPW string address
```

```
27Ø9 CB6F               BIT     5,A             ;If not string, then error
27ØB CAE127             JP      Z,PRMERR
27ØE FDCBØ35E           BIT     3,(IY+3)        ;Can't do if not hard
2712 CAE127             JP      Z,PRMERR
2715 CD2D27             CALL    GETMPW          ;Get and hash the entry
2718 C2C227             JP      NZ,IOERR
271B ØEØØ               LD      C,Ø             ;Init to drive requested
271C           DRIVE    EQU     $-1
271D CDEE27             CALL    GATRD           ;Read GAT into BUF1
272Ø C2C227             JP      NZ,IOERR        ;Back on error
2723 22CE2A             LD      (BUF1+ØCEH),HL  ;Stuff PW
2726 CDEF27             CALL    GATWR           ;Write sector Ø from buf
2729 C2C227             JP      NZ,IOERR        ;Jump on write error
272C C9                 RET                     ;Finished with Repair
               ;
               ;        Enter SYS2 & hash the password
               ;
272D CD3427    GETMPW   CALL    GMPW1           ;Get MPW into buffer
273Ø CØ                 RET     NZ
2731 3EE4               LD      A,ØE4H          ;Hash password (DE) to HL
2733 EF                 RST     28H             ;Ret to what called
               ;
               ;        Place entered password into buffer
               ;
2734 215729    GMPW1    LD      HL,PSWDBUF      ;Point to buffer
2737 E5                 PUSH    HL
2738 Ø6Ø8               LD      B,8             ;Init for 8 chars
273A 1A        GMPW2    LD      A,(DE)          ;P/u a char
273B FEØD               CP      CR              ;End of line?
273D 28ØF               JR      Z,GMPW4
273F FE2C               CP      ','             ;Comma separator?
2741 28ØB               JR      Z,GMPW4
2743 FE22               CP      '"'             ;Closing quote?
2745 28Ø7               JR      Z,GMPW4
2747 13                 INC     DE              ;Bump input pointer
2748 77                 LD      (HL),A          ;Transfer character
2749 23                 INC     HL              ;Bump output pointer
274A 1ØEE               DJNZ    GMPW2           ;Loop until done
274C 18Ø5               JR      CKMPW
274E 362Ø      GMPW4    LD      (HL),' '        ;Buffer with
275Ø 23                 INC     HL              ;  trailing spaces
2751 1ØFB               DJNZ    GMPW4
               ;
               ;        Convert to upper case and check validity
               ;
2753 E1        CKMPW    POP     HL              ;Recover buffer start
2754 E5                 PUSH    HL
2755 Ø6Ø8               LD      B,8
2757 7E                 LD      A,(HL)          ;P/u 1st char
2758 18ØE               JR      CKMPW2          ;  & check <A-Z>
275A 23        CKMPW1   INC     HL
275B 7E                 LD      A,(HL)
275C FE2Ø               CP      ' '             ;Got to a space?
275E 2823               JR      Z,CKMPW7
276Ø FE3Ø               CP      'Ø'             ;Less than 'Ø' is error
2762 3823               JR      C,INVMPW
2764 FE3A               CP      '9'+1           ;<Ø-9> is okay for 2-n
2766 3812               JR      C,CKMPW3
2768 FE41      CKMPW2   CP      'A'             ;Less than "A" is error
276A 381B               JR      C,INVMPW
276C FE5B               CP      'Z'+1           ;<A-Z> is okay
```

```
276E 38ØA      Ø235Ø         JR      C,CKMPW3
277Ø FE61      Ø236Ø         CP      'a'              ;<a-z> convert to
2772 3813      Ø237Ø         JR      C,INVMPW
2774 FE7B      Ø238Ø         CP      'z'+1
2776 3ØØF      Ø239Ø         JR      NC,INVMPW
2778 CBAE      Ø24ØØ         RES     5,(HL)           ;  upper case
277A 1ØDE      Ø241Ø CKMPW3  DJNZ    CKMPW1
277C D1        Ø242Ø CKMPW4  POP     DE               ;Point to buffer start
277D AF        Ø243Ø         XOR     A
277E C9        Ø244Ø         RET
277F 23        Ø245Ø CKMPW5  INC     HL
278Ø BE        Ø246Ø         CP      (HL)             ;No imbedded spaces
2781 2ØØ4      Ø247Ø         JR      NZ,INVMPW
2783 1ØFA      Ø248Ø CKMPW7  DJNZ    CKMPW5
2785 18F5      Ø249Ø         JR      CKMPW4
2787 211C29    Ø25ØØ INVMPW  LD      HL,BADMPW$       ;Init "Invalid PW
278A 3E3F      Ø251Ø         LD      A,63             ;Set extended error
278C B7        Ø252Ø         OR      A                ;Set NZ condition
278D D1        Ø253Ø         POP     DE               ;Clean up stack
278E C9        Ø254Ø         RET
               Ø255Ø ;
               Ø256Ø ;       Reset any file open bits
               Ø257Ø ;
278F E5        Ø258Ø UNOPEN  PUSH    HL               ;Save buffer posn
279Ø C5        Ø259Ø         PUSH    BC
2791 Ø6Ø8      Ø26ØØ         LD      B,8              ;8 entries
2793 2C        Ø261Ø         INC     L                ;Dir + 1
2794 CBAE      Ø262Ø ZAP     RES     5,(HL)           ;Clear file open bit
2796 3E2Ø      Ø263Ø         LD      A,32
2798 85        Ø264Ø         ADD     A,L              ;Pt to next Dir+1
2799 6F        Ø265Ø         LD      L,A
279A 1ØF8      Ø266Ø         DJNZ    ZAP              ;Do 8 entries per direc
279C C1        Ø267Ø         POP     BC
279D E1        Ø268Ø         POP     HL
279E C9        Ø269Ø         RET
               Ø27ØØ ;
279F           Ø271Ø $DSPLY  @@DSPLY                  ;Display a line
               ØØØ14         IFEQ    ØØH,1
               ØØØ15         LD      HL,
               ØØØ16         ENDIF
279F 3EØA      ØØØ17         LD      A,1Ø
27A1 EF        ØØØ18         RST     4Ø
27A2 C8        Ø272Ø         RET     Z
27A3 181D      Ø273Ø         JR      IOERR
               Ø274Ø ;
27A5           Ø275Ø WRSYS   @@WRSSC                  ;Write the sector
27A5 3E36      ØØØ19         LD      A,54
27A7 EF        ØØØ2Ø         RST     4Ø
27A8 2Ø18      Ø276Ø         JR      NZ,IOERR
27AA           Ø277Ø         @@VRSEC                  ;Verify it
27AA 3E32      ØØØ21         LD      A,5Ø
27AC EF        ØØØ22         RST     4Ø
27AD FEØ6      Ø278Ø         CP      6                ;Must be SYSTEM sector
27AF C8        Ø279Ø         RET     Z
27BØ 181Ø      Ø28ØØ         JR      IOERR
               Ø281Ø ;
27B2           Ø282Ø WRSEC   @@WRSEC                  ;Write normal sector
27B2 3E35      ØØØ23         LD      A,53
27B4 EF        ØØØ24         RST     4Ø
27B5 C8        Ø283Ø         RET     Z
27B6 18ØA      Ø284Ø         JR      IOERR
```

```
                 ;
                 ;        Sector read routine
                 ;
27B8 21ØØ2A      RDSEC    LD      HL,BUF1         ;Read sector
27BB                      @@RDSEC
27BB 3E31                 LD      A,49
27BD EF                   RST     4Ø
27BE C8                   RET     Z
27BF FEØ6                 CP      6
27C1 C8                   RET     Z               ;Fall thru to error?
                 ;
                 ;        Error exits
                 ;
27C2 FE3F        IOERR    CP      63              ;Extended error?
27C4 281F                 JR      Z,EXTERR        ;Log it and quit
27C6 26ØØ                 LD      H,Ø             ;Error to HL
27C8 6F                   LD      L,A
27C9 E5                   PUSH    HL              ;Save error code
27CA F6CØ                 OR      ØCØH            ;Set short, return
27CC 4F                   LD      C,A             ;Error to C for
27CD                      @@ERROR                 ;  display
27CD 3E1A                 LD      A,26
27CF EF                   RST     4Ø
                 ;
27DØ 21E828               LD      HL,ABTJOB$      ;Init"Job aborted
                 ;
27D3                      @@LOGOT                 ;Log the msg
                          IFEQ    ØØH,1
                          LD      HL,
                          ENDIF
27D3 3EØC                 LD      A,12
27D5 EF                   RST     4Ø
27D6 E1                   POP     HL              ;Recover error code
27D7 1812                 JR      QUIT$$
                 ;
                 ;        Internal error handler
                 ;
27D9 213429      NIXHARD  LD      HL,NIXHARD$     ;"Can't to hard drive
27DC DD                   DB      ØDDH
27DD 21F728      NOTØ     LD      HL,NOTØ$        ;"Can't do drive Ø
27EØ DD                   DB      ØDDH
27E1 21ØC29      PRMERR   LD      HL,PRMERR$      ;"Parm error
27E4 DD                   DB      ØDDH
27E5             EXTERR   @@LOGOT                 ;Display the error
                          IFEQ    ØØH,1
                          LD      HL,
                          ENDIF
27E5 3EØC                 LD      A,12
27E7 EF                   RST     4Ø
27E8 21FFFF               LD      HL,-1           ;Set abort code
27EB C31B26      QUIT$$   JP      QUIT$
                 ;
                 ;        Read the granule allocation table
                 ;
27EE F6          GATRD    DB      ØF6H            ;Set NZ for test
27EF AF          GATWR    XOR     A               ;Set Z for test
27FØ E5                   PUSH    HL
27F1 F5                   PUSH    AF
27F2 FD56Ø9               LD      D,(IY+9)        ;Dir cylinder
27F5 21ØØ2A               LD      HL,BUF1
27F8 5D                   LD      E,L             ;Set to sector Ø
```

```
27F9 F1                POP     AF
27FA 2807              JR      Z,GATRW1        ;Go if write
27FC                   @@RDSSC
27FC 3E55              LD      A,85
27FE EF                RST     40
27FF 3E14              LD      A,14H
2801 180C              JR      GATRW3
2803         GATRW1    @@WRSSC
2803 3E36              LD      A,54
2805 EF                RST     40
2806 2003              JR      NZ,GATRW2       ;Skip verify if error
2808                   @@VRSEC                 ;Verify the write
2808 3E32              LD      A,50
280A EF                RST     40
280B FE06    GATRW2    CP      6               ;Expect error 6
280D 3E15              LD      A,15H           ;Init "Gat error
280F E1      GATRW3    POP     HL
2810 C9                RET
             ;
             ;         Routine to check on floppy present
             ;
2811 3E28    CKDRV     LD      A,40            ;@DCSTAT
2813 EF                RST     28H
2814 202F              JR      NZ,ILLEG
2816 3E2C              LD      A,44            ;@RSTORE
2818 EF                RST     28H
2819 21002A            LD      HL,BUF1         ;Set up for
281C C5                PUSH    BC              ;  mini ckdrv
281D                   @@TIME                  ;P/u timer ptr
281D 3E13              LD      A,19
281F EF                RST     40
2820 C1                POP     BC
2821 EB                EX      DE,HL           ;Pt HL to
2822 2B                DEC     HL              ;  heartbeat counter
2823 3E2F              LD      A,47            ;@RSLCT
2825 EF                RST     28H             ;Wait till ready
2826 7E                LD      A,(HL)          ;Get heartbeat count
2827 C614              ADD     A,20            ;Init to + 500ms
2829 57                LD      D,A             ;Store for timeout check
282A CD3A28  CK1       CALL    INDEX
282D 20FB              JR      NZ,CK1          ;Get no pulse
282F CD3A28  CK2       CALL    INDEX
2832 28FB              JR      Z,CK2           ;Get pulse
2834 CD3A28  CK3       CALL    INDEX
2837 20FB              JR      NZ,CK3          ;Get no pulse
2839 C9                RET
             ;
283A 7E      INDEX     LD      A,(HL)          ;Get time
283B BA                CP      D               ;Interval expired?
283C 2806              JR      Z,ILLG1
283E 3E2F              LD      A,47            ;@RSLCT
2840 EF                RST     28H
2841 CB4F              BIT     1,A             ;Test for index pulse
2843 C9                RET
             ;
2844 E1      ILLG1     POP     HL              ;Fix stack
2845 3E20    ILLEG     LD      A,32            ;'illegal drv #'
2847 C3C227            JP      IOERR
             ;
             ;
             ;         Messages
```

```
                03850 ;
284A 52         03860 HELLO$  DB      'REPAIR'
     45 50 41 49 52
2850            03870 *GET    CLIENT:3
                03950 ;CLIENTS/ASM - File to establish sign-on headers
                03960 ;
2850 20         03970         DB      ' - 6.2.0 - Copyright 1982/83/84 by Logical'
     2D 20 36 2E 32 2E 30 20
     2D 20 43 6F 70 79 72 69
     67 68 74 20 31 39 38 32
     2F 38 33 2F 38 34 20 62
     79 20 4C 6F 67 69 63 61
     6C
287A 20         03980         DB      ' Systems, Inc.      ',10
     53 79 73 74 65 6D 73 2C
     20 49 6E 63 2E 20 20 20
     20 20 20 0A
                03990 ;
288F 41         04000         DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 20 52 69 67 68 74
     73 20 52 65 73 65 72 76
     65 64 2E 20 4C 69 63 65
     6E 73 65 64 20 31 39 38
     32 2F 38 33 2F 38 34
28B7 20         04010         DB      ' to xxxxxxxxxxxxxxxxxx',10,13
     74 6F 20 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 0A 0D
                03880 ;
28CF 52         03890 ALCAO$  DB      'Repair function complete',CR
     65 70 61 69 72 20 66 75
     6E 63 74 69 6F 6E 20 63
     6F 6D 70 6C 65 74 65 0D
28E8 52         03900 ABTJOB$ DB      'REPAIR aborted',CR
     45 50 41 49 52 20 61 62
     6F 72 74 65 64 0D
28F7 43         03910 NOT0$   DB      'Can''t REPAIR drive 0',CR
     61 6E 27 74 20 52 45 50
     41 49 52 20 64 72 69 76
     65 20 30 0D
290C 50         03920 PRMERR$ DB      'Parameter error',CR
     61 72 61 6D 65 74 65 72
     20 65 72 72 6F 72 0D
291C 49         03930 BADMPW$ DB      'Invalid master password',CR
     6E 76 61 6C 69 64 20 6D
     61 73 74 65 72 20 70 61
     73 73 77 6F 72 64 0D
2934 43         03940 NIXHARD$        DB      'Can''t repair a hard drive',CR
     61 6E 27 74 20 72 65 70
     61 69 72 20 61 20 68 61
     72 64 20 64 72 69 76 65
     0D
                03950 ;
294E 80         03960 PRMTBL$ DB      80H
0020            03970 STR     EQU     20H
294F 23         03980         DB      STR!3,'MPW'
     4D 50 57
2953 00         03990 MRSP    DB      0
2954 0727       04000         DW      MPARM+1
2956 00         04010         NOP
                04020 ;
```

```
ØØØ8          Ø4Ø3Ø PSWDBUF DS      8               ;Password buffer
ØØØ4          Ø4Ø4Ø HASHBUF DS      4               ;Owner & user hashes
ØØ2Ø          Ø4Ø5Ø FCB     DS      32
2AØØ          Ø4Ø6Ø         ORG     $<-8+1<+8
Ø1ØØ          Ø4Ø7Ø BUF1    DS      256
              Ø4Ø8Ø ;
26ØØ          Ø4Ø9Ø         END     BEGIN
```

```
$DSPLY          279F $EXIT          2618 @@1            ØØØØ
@@2             ØØØØ @@3            ØØØØ @@4            ØØØØ
@MOD2           ØØØØ @MOD4          FFFF ABB            ØØ1Ø
ABTJOB$         28E8 ALCAO$         28CF ALIEN1         26A4
ALIEN2          26AB ALIEN3         26EØ ALIEN4         26FA
BADMPW$         291C BEGIN          26ØØ BEGINA         26Ø9
BLNKMPW         4296 BUF1           2AØØ CK1            282A
CK2             282F CK3            2834 CKDRV          2811
CKMPW           2753 CKMPW1         275A CKMPW2         2768
CKMPW3          277A CKMPW4         277C CKMPW5         277F
CKMPW7          2783 CR             ØØØD DRIVE          271C
EXTERR          27E5 FCB            2963 FLAG           ØØ4Ø
GATRD           27EE GATRW1         28Ø3 GATRW2         28ØB
GATRW3          28ØF GATWR          27EF GETMPW         272D
GMPW1           2734 GMPW2          273A GMPW4          274E
HASHBUF         295F HELLO$         284A ILLEG          2845
ILLG1           2844 INDEX          283A INVMPW         2787
IOERR           27C2 LC             26AØ LF             ØØØA
MPARM           27Ø6 MRSP           2953 NIXHARD        27D9
NIXHARD$        2934 NOTØ           27DD NOTØ$          28F7
PGRM            2622 PRMERR         27E1 PRMERR$        29ØC
PRMTBL$         294E PSWDBUF        2957 QUIT$          261B
QUIT$$          27EB RDSEC          27B8 STACK          261C
STR             ØØ2Ø UNOPEN         278F WRSEC          27B2
WRSYS           27A5 ZAP            2794 @@ABORT        8EE9
@@ADTSK         8F7C @@BANK         9494 @@BKSP         9174
@@BREAK         94AA @@CHNIO        8ED4 @@CKBRKC       94F8
@@CKDRV         8FDØ @@CKEOF        9189 @@CKTSK        8F67
@@CLOSE         915F @@CLS          94E2 @@CMNDI        8F13
@@CMNDR         8F28 @@CTL          8D38 @@DATE         8EAA
@@DCSTAT        9ØØF @@DEBUG        8F52 @@DECHEX       9414
@@DIRRD         9381 @@DIRWR        9396 @@DIV16        93FF
@@DIV8          93EA @@DODIR        8FE5 @@DSP          8CFC
@@DSPLY         8D9C @@ERROR        8F3D @@EXIT         8EFE
@@FEXT          92EE @@FLAGS        947E @@FNAME        93Ø3
@@FSPEC         92D9 @@GATRD        936C @@GATWR        93AB
@@GET           8D1Ø @@GTDCB        932D @@GTDCT        9318
@@GTMOD         9342 @@HDFMT        9ØB7 @@HEX16        9453
@@HEX8          943E @@HEXDEC       9429 @@HIGH$        9468
@@INIT          9135 @@KBD          8D74 @@KEY          8CE8
@@KEYIN         8D88 @@KLTSK        8FBB @@LOAD         92AF
@@LOC           919E @@LOF          91B3 @@LOGER        8DD3
@@LOGOT         8DE8 @@MSG          8E1F @@MUL16        93D5
@@MUL8          93CØ @@OPEN         914A @@PARAM        8E95
@@PAUSE         8E8Ø @@PEOF         91C8 @@POSN         91DD
@@PRINT         8E34 @@PRT          8D4C @@PUT          8D24
@@RAMDIR        8FFA @@RDSEC        9Ø8D @@RDSSC        9357
@@READ          91F2 @@REMOV        912Ø @@RENAM        91ØB
@@REW           92Ø7 @@RMTSK        8F91 @@RPTSK        8FA6
@@RREAD         921C @@RSLCT        9Ø78 @@RSTOR        9Ø39
@@RUN           92C4 @@RWRIT        9231 @@SEEK         9Ø63
@@SEEKSC        9246 @@SKIP         925B @@SLCT         9Ø24
@@STEPI         9Ø4E @@TIME         8EBF @@VDCTL        8E6B
@@VER           927Ø @@VRSEC        9ØA2 @@WEOF         9285
@@WHERE         8D6Ø @@WRITE        929A @@WRSEC        9ØCC
@@WRSSC         9ØE1 @@WRTRK        9ØF6
26ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

TAPE1ØØ/CMD - Read or write a Model 1ØØ tape

Tape1ØØ allows cassette tapes written on a Model 1ØØ to be read in and save as a disk file, and vice versa.

```
                 ;TAPE1ØØ - Tape/Disk & Disk/Tape Xfer Utility
ØØØØ                     TITLE   <TAPE1ØØ - LS-DOS 6.2>
                 ;
F44Ø             BREAKLC EQU     ØF44ØH          ;<BREAK> key location
ØØ3A             LOADA   EQU     3AH             ; LD  A,(nnnn) opcode
ØØ16             WRMASK  EQU     'W'-'A'         ;WRINTMASK port mask byte
ØØØC             MODMASK EQU     'M'-'A'         ;MODOUT port mask byte
                 ;
ØØ3A             @INIT   EQU     58              ;@INIT SVC #
ØØ3B             @OPEN   EQU     59              ;@OPEN SVC #
                 ;
ØØEØ             PORTEØ  EQU     ØEØH
ØØEC             MODOUT  EQU     ØECH
ØØFF             PORTFF  EQU     ØFFH
ØØ78             OPREG$  EQU     78H             ;Operating Register
ØØ84             @OPREG  EQU     84H             ;Video/Keyboard Control Port
F8ØØ             VIDEO   EQU     ØF8ØØH          ;Start of Video RAM
                 ;
ØØ22             WHICH1  EQU     22H             ;Which one - Ø or 1 ?
ØØØF             TOOSHRT EQU     ØFH             ;Pulse too Short ?
ØØ3E             TOOLONG EQU     3EH             ;Pulse too Long ?
ØØØ6             ROUTOFF EQU     6               ;Interrupt rout offset
ØØØD             DIFFER  EQU     ØDH             ;Difference between 2 pulses
2B2F             DELAYØ  EQU     2B2FH           ;Bit = Ø Delay count
1217             DELAY1  EQU     1217H           ;Bit = 1 Delay count
                 ;
ØØØE             CURON   EQU     14              ;Cursor on
ØØØF             CUROFF  EQU     15              ;Cursor off
                 ;
ØØØØ             *GET    SVCMAC:3                ;SVC Macro equivalents
                 ;SVCMAC/ASM - LS-DOS Version VI
                 *LIST   OFF
                 *LIST   ON
ØØØØ             *GET    VALUES:3                ;Misc. equates
                 ;VALUES/ASM - Version 6
                 *LIST OFF
                 *LIST ON
ØØØØ             *GET    COPYCOM:3               ;Copyright message
                 ; COPYCOM - File for Copyright COMment block
                 ;
ØØØØ                     COM     '<*(C) 1982,83,84 by LSI*>'
                 ;
26ØØ                     ORG     26ØØH
                 ;
                 START
26ØØ                     @@CKBRKC                ;Check for break
26ØØ 3E6A                LD      A,1Ø6
26Ø2 EF                  RST     4Ø
26Ø3 28Ø4                JR      Z,STARTA        ;Continue if not
26Ø5 21FFFF              LD      HL,-1           ;  else abort
26Ø8 C9                  RET
                 ;
26Ø9 ED735C27    STARTA  LD      (OLDSP+1),SP    ;Save entry stack
26ØD CDC727              CALL    DOINIT          ;Do initialization
                 ;
                 ;       Was READ or WRITE entered ?
                 ;
261Ø 3AD729              LD      A,(RRESP)       ;P/u read response
2613 47                  LD      B,A             ;Xfer to B
2614 3ACF29              LD      A,(WRESP)       ;P/u write response
2617 A8                  XOR     B               ;Are both the same ?
```

```
2618 2807      00600          JR      Z,INP_R_W       ;Yes - prompt
               00610 ;
               00620 ;        Both weren't entered - which one was
               00630 ;
261A 04        00640 CHKPRM   INC     B               ;READ entered ?
261B 05        00650          DEC     B
261C 281C      00660          JR      Z,WRTAPE        ;<W>rite a tapefile
261E C3B126    00670          JP      RDTAPE          ;<R>ead a tapefile
               00680 ;
               00690 ;        Prompt for READ or WRITE
               00700 ;
2621 E5        00710 INP_R_W  PUSH    HL              ;Save command ptr
               00720 ;
2622 21DB28    00730          LD      HL,RDORWR       ;"Read or Write"
2625 CD4928    00740          CALL    DSPLY
               00750 ;
               00760 ;        Input R (Read) or W (Write)
               00770 ;
2628 0601      00780          LD      B,1             ;Take input, 1 char
262A CD3528    00790          CALL    INPUT
262D 7E        00800          LD      A,(HL)          ;P/u first char
262E E1        00810          POP     HL              ;Recover command ptr
262F CBAF      00820          RES     5,A             ;Convert to U/C
2631 FE52      00830          CP      'R'             ;<R>ead ?
2633 CAB126    00840          JP      Z,RDTAPE
2636 FE57      00850          CP      'W'             ;<W>rite ?
2638 20E7      00860          JR      NZ,INP_R_W      ;No - re-prompt
               00870 ;
               00880 ;        WRITE diskfile to tapefile
               00890 ;
263A 11F12D    00900 WRTAPE   LD      DE,FCB1         ;DE => Source FCB
263D           00910          @@FSPEC                 ;If a bad spec,
263D 3E4E      00003          LD      A,78
263F EF        00004          RST     40
2640 C40828    00920          CALL    NZ,PRSOUR       ;  prompt for source
               00930 ;
               00940 ;        WRITE - check if destination filespec input
               00950 ;
2643 11112E    00960 WRTAPE2  LD      DE,FCB2         ;DE => Destination FCB
2646           00970          @@FSPEC
2646 3E4E      00005          LD      A,78
2648 EF        00006          RST     40
2649 C41028    00980          CALL    NZ,PRDEST       ;Prompt for destination
264C CDAC27    00990          CALL    GTFILE          ;Xfer into Filename
               01000 ;
               01010 ;        Open Disk Source file
               01020 ;
264F 11F12D    01030 OPDSRC   LD      DE,FCB1         ;DE => Source
2652 CDCF2C    01040          CALL    OPEN
2655 C24627    01050          JP      NZ,IOERR        ;NZ - abort
               01060 ;
               01070 ;        Can this disk file fit into memory ?
               01080 ;
2658 2AFD2D    01090          LD      HL,(FCB1+12)    ;P/u ERN
265B 24        01100          INC     H               ;Too big ?
265C 25        01110          DEC     H
265D C2BF29    01120          JP      NZ,TOOBIG       ;Yes - forget it
2660 3E00      01130 ENUF     LD      A,$-$           ;Enough memory ?
2662 C630      01140          ADD     A,MEM<-8        ;Add mem start
2664 BD        01150          CP      L
2665 DABF29    01160          JP      C,TOOBIG        ;No - forget it
```

```
             Ø117Ø ;
             Ø118Ø ;        Read in Disk file & Write to tape
             Ø119Ø ;
2668 CD3B2A  Ø12ØØ          CALL    PRTAPE          ;Display "Ready Tape"
266B CDBC2C  Ø121Ø          CALL    CURSOFF         ;Turn of cursor
266E 217B27  Ø122Ø          LD      HL,READING      ;Init "Reading : "
2671 CD4928  Ø123Ø          CALL    DSPLY           ;Display line
2674 219D27  Ø124Ø          LD      HL,DFBUF        ;HL => Disk Filename
2677 CD4928  Ø125Ø          CALL    DSPLY
267A CD692D  Ø126Ø          CALL    READSRC         ;Read the source file
267D CDCA2D  Ø127Ø          CALL    GETPOS          ;Get new cursor pos
268Ø CD9B2D  Ø128Ø          CALL    ENDOKI          ;Bring in Video
2683 218527  Ø129Ø          LD      HL,WRITING      ;Display "Writing : "
2686 CD8Ø2C  Ø13ØØ          CALL    DISPSTR
2689 219Ø27  Ø131Ø          LD      HL,FILENM       ;"filenm"
268C CD8Ø2C  Ø132Ø          CALL    DISPSTR
268F CD1A2A  Ø133Ø          CALL    CASSON          ;Turn on cassette
2692 Ø68Ø    Ø134Ø          LD      B,8ØH           ;Pause a bit
2694         Ø135Ø          @@PAUSE
2694 3E1Ø    ØØØØ7          LD      A,16
2696 EF      ØØØØ8          RST     4Ø
2697 CDEC2B  Ø136Ø          CALL    WRHEAD          ;Write Header
269A CD192C  Ø137Ø          CALL    WRDAT           ;Write Data
269D 2A6627  Ø138Ø          LD      HL,(CURPOS)     ;P/u new cursor pos
26AØ CDE32D  Ø139Ø          CALL    GETCRS
26A3 Ø6Ø3    Ø14ØØ          LD      B,3             ;Give to system
26A5         Ø141Ø          @@VDCTL
26A5 3EØF    ØØØØ9          LD      A,15
26A7 EF      ØØØ1Ø          RST     4Ø
26A8 CDA92D  Ø142Ø          CALL    DISDOKI         ;Restore video
26AB CD2B2A  Ø143Ø          CALL    CASSOFF         ;Turn off cassette
26AE C35827  Ø144Ø          JP      EXIT            ;Clean exit
             Ø145Ø ;
             Ø146Ø ;        Get Source & Destination for READ
             Ø147Ø ;
26B1 11F12D  Ø148Ø RDTAPE   LD      DE,FCB1         ;First filespec legal ?
26B4         Ø149Ø          @@FSPEC
26B4 3E4E    ØØØ11          LD      A,78
26B6 EF      ØØØ12          RST     4Ø
26B7 28ØD    Ø15ØØ          JR      Z,CHKSEC        ;Yes - check for second
             Ø151Ø ;
             Ø152Ø ;        Accept first filename on tape
             Ø153Ø ;
26B9 3EC9    Ø154Ø          LD      A,ØC9H
26BB 32ØD2B  Ø155Ø          LD      (CORRECT),A
26BE 11112E  Ø156Ø          LD      DE,FCB2         ;Prompt for dest filename
26C1 CD2D28  Ø157Ø          CALL    PRDEST2         ;Prompt for dest
26C4 1825    Ø158Ø          JR      READFIL         ;  and read file
             Ø159Ø ;
             Ø16ØØ ;        Copy source FCB into destination
             Ø161Ø ;
26C6 E5      Ø162Ø CHKSEC   PUSH    HL              ;Save comm ptr
26C7 EB      Ø163Ø          EX      DE,HL
26C8 11112E  Ø164Ø          LD      DE,FCB2         ;DE => Disk FCB
26CB Ø12ØØØ  Ø165Ø          LD      BC,32
26CE D5      Ø166Ø          PUSH    DE              ;Save dest FCB
26CF EDBØ    Ø167Ø          LDIR
26D1 D1      Ø168Ø          POP     DE
26D2 E1      Ø169Ø          POP     HL
             Ø17ØØ ;
             Ø171Ø ;        P/u destination filespec
```

```
                ;
26D3 2B                 DEC     HL              ;Skip leading spaces
26D4 23         SKPSPC  INC     HL
26D5 7E                 LD      A,(HL)          ;P/u char
26D6 FE2Ø               CP      ' '             ;Space ?
26D8 28FA               JR      Z,SKPSPC
26DA FEØE               CP      CR+1            ;Eol ?
26DC 38Ø7               JR      C,GTFILE2       ;Yes - use default
26DE FE28               CP      '('             ;Eol ?
26EØ 28Ø3               JR      Z,GTFILE2
26E2                    @@FSPEC                 ;Xfer in if legal
26E2 3E4E               LD      A,78
26E4 EF                 RST     4Ø
                ;
                ;       Transfer filename into buffer left just'd
                ;
26E5 11F12D     GTFILE2 LD      DE,FCB1         ;DE => Source
26E8 CDAC27             CALL    GTFILE          ;Stuff Filename into buff
                ;
                ;       Read in Tape Source file
                ;
26EB 11112E     READFIL LD      DE,FCB2         ;@INIT the dest file
26EE CDCB2C             CALL    INIT
26F1 C24627             JP      NZ,IOERR
26F4 DD4EØ6             LD      C,(IX+6)        ;P/u drive #
26F7                    @@CKDRV                 ;Write protected ?
26F7 3E21               LD      A,33
26F9 EF                 RST     4Ø
26FA 3EØF               LD      A,15            ;Write Protected Disk
26FC DA4627             JP      C,IOERR         ;Good bye
26FF CD3B2A             CALL    PRTAPE          ;"Ready Cassette"
27Ø2 CDBC2C             CALL    CURSOFF
27Ø5 CD9B2D             CALL    ENDOKI          ;Bring in KI & DO RAM
27Ø8 CDCA2D             CALL    GETPOS          ;Calculate cursor posn
27ØB 217B27             LD      HL,READING      ;Display "Reading : "
27ØE CD8Ø2C             CALL    DISPSTR
2711 CD1A2A             CALL    CASSON          ;Turn on cassette
2714 CDD82A             CALL    RDHEAD          ;Search for header
2717 CD492A             CALL    RDDAT           ;Read in Data
271A F3                 DI                      ;Make sure off
271B CD2B2A             CALL    CASSOFF         ;Turn off cassette
271E 218527             LD      HL,WRITING      ;Display "Writing : "
2721 CD8Ø2C             CALL    DISPSTR         ;
2724 219D27             LD      HL,DFBUF        ;HL => Destination
2727 CD8Ø2C             CALL    DISPSTR         ;
272A 2A6627             LD      HL,(CURPOS)     ;P/u new cursor position
272D CDE32D             CALL    GETCRS          ;Convert to Row, Column
273Ø Ø6Ø3               LD      B,3             ;Give system new cursor
2732                    @@VDCTL                 ;
2732 3EØF               LD      A,15
2734 EF                 RST     4Ø
2735 CDA92D             CALL    DISDOKI         ;Enable real RAM
2738 18Ø6               JR      WRTDES2         ;
273A CDA92D     FORNOW  CALL    DISDOKI         ;Enable real RAM
273D CD2B2A             CALL    CASSOFF         ;Turn off cassette
274Ø CD512D     WRTDES2 CALL    WRTDEST         ;Write Destination file
2743 C35827             JP      EXIT            ;Clean exit
                ;
                ;
2746 6F         IOERR   LD      L,A             ;Xfer error # to HL
2747 26ØØ               LD      H,Ø             ;
```

```
2749 F6CØ                   OR      ØCØH              ;Abbrev, return
274B 4F                     LD      C,A
274C                        @@ERROR                   ;Display error
274C 3E1A                   LD      A,26
274E EF                     RST     4Ø
274F 18ØA                   JR      OLDSP             ;  and abort
                 ;
2751 C35427      ILLEGAL    JP      ABORT             ;For now
                 ;
2754 21FFFF      ABORT      LD      HL,-1             ;Show error return
2757 DD                     DB      ØDDH              ;Skip LD HL,Ø
2758 21ØØØØ      EXIT       LD      HL,Ø              ;Clean exit
275B 31ØØØØ      OLDSP      LD      SP,$-$            ;P/u original SP
275E FB                     EI                        ;Re-enable interrupts
275F                        @@CKBRKC                  ;Clear Break
275F 3E6A                   LD      A,1Ø6
2761 EF                     RST     4Ø
2762 C9                     RET                       ;  and RETurn
                 ;
2763 ØØ          DLEN       DB      Ø,Ø,Ø
     ØØ ØØ
2766 ØØØØ        CURPOS     DW      Ø                 ;Cursor Position
2768 ØA          READERR    DB      LF,'Tape Read Error  ',CR
     54 61 7Ø 65 2Ø 52 65 61
     64 2Ø 45 72 72 6F 72 2Ø
     2Ø ØD
277B 52          READING    DB      'Reading: ',ETX
     65 61 64 69 6E 67 3A 2Ø
     Ø3
2785 ØA          WRITING    DB      LF,'Writing: ',ETX
     57 72 69 74 69 6E 67 3A
     2Ø Ø3
279Ø 46          FILENM     DB      'FILENM',CR
     49 4C 45 4E 4D ØD
ØØØ6             BUFFER     DS      6
279D 46          DFBUF      DB      'Filename/ext:d',ETX
     69 6C 65 6E 61 6D 65 2F
     65 78 74 3A 64 Ø3
                 ;
                 ;
                 ;          GTFILE - Stuff filename from FCB into buffer
                 ;          DE => FCB with filename contained
                 ;
27AC 219Ø27      GTFILE     LD      HL,FILENM         ;HL => Filename buffered
27AF E5                     PUSH    HL                ;Save it
27BØ Ø6Ø6                   LD      B,6               ;Init to all spaces
27B2 362Ø        CLEAN      LD      (HL),' '
27B4 23                     INC     HL
27B5 1ØFB                   DJNZ    CLEAN
27B7 E1                     POP     HL                ;HL => Filename dest
27B8 Ø6Ø6                   LD      B,6               ;Only accept first 6
                 ;
27BA 1A          GETFILN    LD      A,(DE)            ;P/u char
27BB FEØE                   CP      CR+1              ;End ?
27BD D8                     RET     C                 ;Yes - done
27BE FE2E                   CP      '.'               ;Start of password?
27CØ C8                     RET     Z                 ;Yes - done
27C1 77                     LD      (HL),A            ;Stuff into filename buff
27C2 23                     INC     HL                ;Bump
27C3 13                     INC     DE
27C4 1ØF4                   DJNZ    GETFILN
```

```
27C6 C9         02730          RET                       ;Done - RETurn
                02740 ;
                02750 ;        DOINIT - Do initialization
                02760 ;
27C7            02770 DOINIT   @@FLAGS                   ;IY => System Flags
27C7 3E65       00023          LD      A,101
27C9 EF         00024          RST     40
                02780 ;
                02790 ;        Calculate highest mem address of buffer
                02800 ;
27CA E5         02810          PUSH    HL                ;Save command line stuff
27CB 210000     02820          LD      HL,0              ;P/u HIGH$
27CE 45         02830          LD      B,L
27CF FDCB024E   02840          BIT     1,(IY+CFLAG$)     ;@CMNDR ?
27D3 2801       02850          JR      Z,USEHI
27D5 04         02860          INC     B                 ;Use LOW$
27D6            02870 USEHI    @@HIGH$
27D6 3E64       00025          LD      A,100
27D8 EF         00026          RST     40
27D9 23         02880          INC     HL                ;Set hi-mem byte
27DA 25         02890          DEC     H                 ;Give some lee-way
27DB 25         02900          DEC     H
27DC 7C         02910          LD      A,H               ;  & stuff in R/W routines
27DD 326126     02920          LD      (ENUF+1),A
                02930 ;
                02940 ;        Display Log-on message
                02950 ;
27E0 215328     02960          LD      HL,HELLO$         ;Display banner
27E3 CD4928     02970          CALL    DSPLY
27E6 E1         02980          POP     HL                ;Process parm line
                02990 ;
                03000 ;        P/u READ or WRITE parm if entered
                03010 ;
27E7 E5         03020          PUSH    HL                ;Save HL
27E8 2B         03030          DEC     HL                ;Back up one
27E9 23         03040 CKPLP    INC     HL                ;Bump
27EA 7E         03050          LD      A,(HL)            ;P/u char
27EB FE0E       03060          CP      CR+1              ;Eol ?
27ED 380D       03070          JR      C,DUNLIN          ;Yes - done
27EF FE28       03080          CP      '('               ;Paramter entered ?
27F1 20F6       03090          JR      NZ,CKPLP          ;No - go til eol
                03100 ;
                03110 ;        Process parameter entry
                03120 ;
27F3 11C829     03130          LD      DE,PARMTBL        ;DE => Param table
27F6            03140          @@PARAM
27F6 3E11       00027          LD      A,17
27F8 EF         00028          RST     40
27F9 C2BB29     03150          JP      NZ,PRMERR         ;NZ - parameter error
27FC E1         03160 DUNLIN   POP     HL                ;Rcvr command ptr
                03170 ;
                03180 ;        If C=N entered then use checksum
                03190 ;
27FD 01FFFF     03200 CPARM    LD      BC,0FFFFH         ;Default no checksum
2800 04         03210          INC     B                 ;User requesting checksum?
2801 C0         03220          RET     NZ                ;Yes, return
2802 3EC9       03230          LD      A,0C9H            ;Init RET opcode
2804 32242B     03240          LD      (CHKERR+1),A      ;Stuff into Checksum error
2807 C9         03250          RET
                03260 ;
                03270 ;        PRSOUR/PRDEST - Prompt for Source & Destination
```

```
               Ø328Ø ;
28Ø8 E5        Ø329Ø PRSOUR  PUSH    HL              ;Save HL
28Ø9 21ØA29    Ø33ØØ         LD      HL,DSF          ;"Disk Source Filename"
28ØC Ø617      Ø331Ø         LD      B,23            ;23 chars max
28ØE 18Ø6      Ø332Ø         JR      DOINPUT
281Ø E5        Ø333Ø PRDEST  PUSH    HL              ;Save HL
2811 212329    Ø334Ø         LD      HL,TDF          ;"Tape Dest Filename"
2814 Ø6Ø6      Ø335Ø         LD      B,6             ;6 char max
2816 CD4928    Ø336Ø DOINPUT CALL    DSPLY           ;Display prompt
2819 E5        Ø337Ø         PUSH    HL              ;Save prompt start
281A CD3528    Ø338Ø         CALL    INPUT           ;Input
281D           Ø339Ø         @@FSPEC                 ;Legal ?
281D 3E4E      ØØØ29         LD      A,78
281F EF        ØØØ3Ø         RST     4Ø
282Ø E1        Ø34ØØ         POP     HL              ;HL => Prompt string
2821 2ØF3      Ø341Ø         JR      NZ,DOINPUT      ;Reprompt on bad name
2823 E1        Ø342Ø         POP     HL              ;Recover ptr
2824 C9        Ø343Ø         RET                     ;  and return
               Ø344Ø ;
               Ø345Ø ;       PRSOUR2/PRDEST2 - Prompt for READ source/dest
               Ø346Ø ;
2825 E5        Ø347Ø PRSOUR2 PUSH    HL              ;Save HL
2826 21F128    Ø348Ø         LD      HL,TSF          ;"Tape Source filename"
2829 Ø6Ø6      Ø349Ø         LD      B,6             ;6 char max
282B 18E9      Ø35ØØ         JR      DOINPUT
282D E5        Ø351Ø PRDEST2 PUSH    HL              ;Save HL
282E 214129    Ø352Ø         LD      HL,DDF          ;"Disk Destination file"
2831 Ø617      Ø353Ø         LD      B,23            ;23 char max
2833 18E1      Ø354Ø         JR      DOINPUT
               Ø355Ø ;
               Ø356Ø ;       INPUT - Line input routine
               Ø357Ø ;
2835 D5        Ø358Ø INPUT   PUSH    DE              ;Save DE
2836 C5        Ø359Ø         PUSH    BC              ;  and BC
2837 21312E    Ø36ØØ         LD      HL,INBUFF       ;HL => Input buffer
283A           Ø361Ø         @@KEYIN                 ;Input line
283A 3EØ9      ØØØ31         LD      A,9
283C EF        ØØØ32         RST     4Ø
283D DA5427    Ø362Ø         JP      C,ABORT         ;<BREAK> abort
284Ø C1        Ø363Ø         POP     BC              ;Restore regs
2841 D1        Ø364Ø         POP     DE
2842 C9        Ø365Ø         RET
               Ø366Ø ;
2843 D5        Ø367Ø DSP     PUSH    DE              ;Save DE
2844           Ø368Ø         @@DSP                   ;Output char
2844 3EØ2      ØØØ33         LD      A,2
2846 EF        ØØØ34         RST     4Ø
2847 18Ø4      Ø369Ø         JR      EXDSP
               Ø37ØØ ;
2849 D5        Ø371Ø DSPLY   PUSH    DE              ;Save DE
284A           Ø372Ø         @@DSPLY                 ;Display message
               ØØØ35         IFEQ    ØØH,1
               ØØØ36         LD      HL,
               ØØØ37         ENDIF
284A 3EØA      ØØØ38         LD      A,1Ø
284C EF        ØØØ39         RST     4Ø
284D D1        Ø373Ø EXDSP   POP     DE              ;Rcvr DE
284E C8        Ø374Ø         RET     Z               ;RETurn if OK
284F C34627    Ø375Ø         JP      IOERR           ;  else abort
               Ø376Ø ;
2852 ØØ        Ø377Ø COUNT   DB      Ø               ;Count
```

```
                 Ø378Ø ;
2853 1C          Ø379Ø HELLO$  DB      1CH,1FH,'TAPE1ØØ'
     1F 54 41 5Ø 45 31 3Ø 3Ø
285C             Ø38ØØ *GET    CLIENT:3
                 Ø424Ø ;CLIENTS/ASM - File to establish sign-on headers
                 Ø425Ø ;
285C 2Ø          Ø426Ø         DB      ' - 6.2.Ø - Copyright 1982/83/84 by Logical'
     2D 2Ø 36 2E 32 2E 3Ø 2Ø
     2D 2Ø 43 6F 7Ø 79 72 69
     67 68 74 2Ø 31 39 38 32
     2F 38 33 2F 38 34 2Ø 62
     79 2Ø 4C 6F 67 69 63 61
     6C
2886 2Ø          Ø427Ø         DB      ' Systems, Inc.      ',1Ø
     53 79 73 74 65 6D 73 2C
     2Ø 49 6E 63 2E 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø ØA
                 Ø428Ø ;
289B 41          Ø429Ø         DB      'All Rights Reserved. Licensed 1982/83/84'
     6C 6C 2Ø 52 69 67 68 74
     73 2Ø 52 65 73 65 72 76
     65 64 2E 2Ø 4C 69 63 65
     6E 73 65 64 2Ø 31 39 38
     32 2F 38 33 2F 38 34
28C3 2Ø          Ø43ØØ         DB      ' to xxxxxxxxxxxxxxxxxx',1Ø,13
     74 6F 2Ø 78 78 78 78 78
     78 78 78 78 78 78 78 78
     78 78 78 78 78 ØA ØD
                 Ø381Ø ;
28DB 3C          Ø382Ø RDORWR  DB      '<R>ead or <W>rite ? ',CURON,ETX
     52 3E 65 61 64 2Ø 6F 72
     2Ø 3C 57 3E 72 69 74 65
     2Ø 3F 2Ø ØE Ø3
28F1 54          Ø383Ø TSF     DB      'Tape Source Filespec ? ',CURON,ETX
     61 7Ø 65 2Ø 53 6F 75 72
     63 65 2Ø 46 69 6C 65 73
     7Ø 65 63 2Ø 3F 2Ø ØE Ø3
29ØA 44          Ø384Ø DSF     DB      'Disk Source Filespec ? ',CURON,ETX
     69 73 6B 2Ø 53 6F 75 72
     63 65 2Ø 46 69 6C 65 73
     7Ø 65 63 2Ø 3F 2Ø ØE Ø3
2923 54          Ø385Ø TDF     DB      'Tape Destination Filespec ? ',CURON,ETX
     61 7Ø 65 2Ø 44 65 73 74
     69 6E 61 74 69 6F 6E 2Ø
     46 69 6C 65 73 7Ø 65 63
     2Ø 3F 2Ø ØE Ø3
2941 44          Ø386Ø DDF     DB      'Disk Destination Filespec ? ',CURON,ETX
     69 73 6B 2Ø 44 65 73 74
     69 6E 61 74 69 6F 6E 2Ø
     46 69 6C 65 73 7Ø 65 63
     2Ø 3F 2Ø ØE Ø3
295F 52          Ø387Ø TREADY  DB      'Ready Cassette & Press <ENTER>'
     65 61 64 79 2Ø 43 61 73
     73 65 74 74 65 2Ø 26 2Ø
     5Ø 72 65 73 73 2Ø 3C 45
     4E 54 45 52 3E
297D ØE          Ø388Ø         DB      CURON,ETX
     Ø3
297F 5Ø          Ø389Ø PRMERR$ DB      'Parameter error',LF,CR
     61 72 61 6D 65 74 65 72
     2Ø 65 72 72 6F 72 ØA ØD
```

```
2990 46         03900 TOOBIG$ DB       'File too large to fit in available '
     69 6C 65 20 74 6F 6F 20
     6C 61 72 67 65 20 74 6F
     20 66 69 74 20 69 6E 20
     61 76 61 69 6C 61 62 6C
     65 20
29B3 6D         03910         DB       'memory',LF,CR
     65 6D 6F 72 79 0A 0D
                03920 ;
                03930 ;
                03940 ;        Error Exit routine
                03950 ;
29BB 217F29     03960 PRMERR  LD       HL,PRMERR$        ;"Parameter Error"
29BE DD         03970         DB       0DDH              ;Skip
29BF 219029     03980 TOOBIG  LD       HL,TOOBIG$        ;"File too Big"
                03990 ;
29C2            04000         @@LOGOT                    ;Display error
                00040         IFEQ     00H,1
                00041         LD       HL,
                00042         ENDIF
29C2 3E0C       00043         LD       A,12
29C4 EF         00044         RST      40
29C5 C35427     04010         JP       ABORT             ;Good bye
                04020 ;
                04030 ;        Parameter Table
                04040 ;
29C8 80         04050 PARMTBL DB       80H               ;6.x @PARAM
29C9 55         04060         DB       FLAG!ABB!5
29CA 57         04070         DB       'WRITE'
     52 49 54 45
29CF 00         04080 WRESP   DB       0
29D0 E629       04090         DW       WPARM
                04100 ;
29D2 54         04110         DB       FLAG!ABB!4
29D3 52         04120         DB       'READ'
     45 41 44
29D7 00         04130 RRESP   DB       0
29D8 E429       04140         DW       RPARM
                04150 ;
29DA 54         04160         DB       FLAG!ABB!4
29DB 43         04170         DB       'CHECK'
     48 45 43 4B
29E0 00         04180 CRESP   DB       0
29E1 FE27       04190         DW       CPARM+1
                04200 ;
29E3 00         04210         DB       0
                04220 ;
29E4 0000       04230 RPARM   DW       0
29E6 0000       04240 WPARM   DW       0
                04250 ;
29E8 00         04260         DC       50,0              ;Patch space
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00
                04270 ;
2A1A            04280 *GET    TAPE100A:3
                04310 ;TAPE100A/ASM - Tape I/O routines
                ;        CASSON - Turn Cassette Motor On
                ;
2A1A F3         CASSON  DI                       ;Disable interrupts
2A1B CDB52D             CALL    SWAP38           ;Grab RST 38H vector
2A1E DBEØ               IN      A,(PORTEØ)       ;Clear any latches
2A2Ø DBEC               IN      A,(MODOUT)       ;Clear any latches
2A22 3EØ2               LD      A,2              ;Motor on, slow speed
2A24 D3EC               OUT     (MODOUT),A       ;Turn on motor
2A26 3EØ3               LD      A,3              ;Disable other interrupts
2A28 D3EØ               OUT     (PORTEØ),A
2A2A C9                 RET
                ;
                ;        CASSOFF - Turn off Cassette Motor
                ;
2A2B FD7E16     CASSOFF LD      A,(IY+WRMASK)    ;P/u original
2A2E D3EØ               OUT     (PORTEØ),A       ;Set up R/F interrupt
2A3Ø DBFF               IN      A,(PORTFF)       ;Clear 15ØØ bd interrupts
2A32 FD7EØC             LD      A,(IY+MODMASK)   ;Turn off motor
2A35 D3EC               OUT     (MODOUT),A
2A37 CDB52D             CALL    SWAP38           ;Restore RST 38H vector
2A3A C9                 RET
                ;
                ;        PRTAPE - Prompt for "Tape Ready" & turn motor on
                ;
2A3B 215F29     PRTAPE  LD      HL,TREADY        ;"Ready cassette & <ENTER>
2A3E CD4928             CALL    DSPLY
2A41 Ø6Ø1       NOTENT  LD      B,1              ;Just 1 char
2A43 CD3528             CALL    INPUT            ;<BREAK> or <ENTER>
2A46 C3BC2C             JP      CURSOFF          ;Turn off Cursor & RETurn
                ;
                ;        RDDAT - Read in a tape file
                ;
2A49 21ØØ2F     RDDAT   LD      HL,MEM-1ØØH      ;HL => Start of file
2A4C 24         RDDAT2  INC     H                ;Bump hi-byte
2A4D CD582A             CALL    RDDATA           ;Read a block
2A5Ø C8                 RET     Z                ;Eof ?
2A51 3EØØ       EOTF    LD      A,$-$            ;At top of memory ?
2A53 BC                 CP      H
2A54 2ØF6               JR      NZ,RDDAT2        ;No
2A56 B7                 OR      A                ;Top of mem -
2A57 C9                 RET                      ;RETurn NZ
                ;
                ;        RDDATA - Read in a block of Data
                ;        HL => Destination of Block
                ;
2A58 CD392B     RDDATA  CALL    RDSYNC           ;Read sync field
2A5B CDA72B             CALL    RDBYTE           ;Read a byte
2A5E FE8D               CP      8DH              ;Legal ?
2A6Ø C25127             JP      NZ,ILLEGAL       ;No - bad news
2A63 11ØØØØ             LD      DE,Ø             ;D=EOF flag, E = checksum
                ;
2A66 CDA72B     RDLP1   CALL    RDBYTE           ;Read a byte
2A69 77                 LD      (HL),A           ;Stuff into buffer
                ;
                ;        Check for End of File byte X'1A'
                ;
2A6A FE1A               CP      1AH              ;Eof ?
2A6C 2ØØ5               JR      NZ,AFTER         ;No
2A6E BA                 CP      D                ;Been here before ?
2A6F 28Ø2               JR      Z,AFTER          ;First time ?
2A71 57                 LD      D,A              ;Set D = 1AH
```

```
2A72 45                 LD      B,L             ;Yes - set B = pos
              ;
              ;         Add byte to checksum
              ;
2A73 83       AFTER     ADD     A,E             ;Add checksum
2A74 5F                 LD      E,A             ;Xfer back to E
2A75 2C                 INC     L               ;Bump
2A76 2ØEE               JR      NZ,RDLP1
2A78 ED44               NEG                     ;Negate checksum
2A7A 5F                 LD      E,A             ;Stuff back in E
              ;
              ;         Verify Checksum byte
              ;
2A7B CDA72B             CALL    RDBYTE          ;Read in byte
2A7E BB                 CP      E               ;Checksums match ?
2A7F C4232B             CALL    NZ,CHKERR       ;No - checksum error
              ;
              ;         Stuff EOF offset byte into WRTDEST routine
              ;
2A82 7C                 LD      A,H             ;P/u eom
2A83 325C2D             LD      (EOTF2+1),A     ;Stuff into WRTDEST
2A86 78                 LD      A,B             ;P/u byte
2A87 3C                 INC     A               ;Bump
2A88 32612D             LD      (OFFSET+1),A
              ;
              ;         Read past 2Ø dummy zeroes
              ;
2A8B Ø614               LD      B,2Ø
2A8D CDA72B   RDLP2     CALL    RDBYTE
2A9Ø 1ØFB               DJNZ    RDLP2
              ;
              ;         Set Z flag if at EOF
              ;
2A92 7A                 LD      A,D             ;Eof ?
2A93 FE1A               CP      1AH
2A95 C9                 RET                     ;Done
              ;
              ;         RDBIT - Read a Bit from Cassette
              ;
2A96 ØEØØ     RDBIT     LD      C,Ø             ;Init count = Ø
2A98 FB                 EI                      ;Back on
2A99 ØC       RBLP      INC     C               ;Bump count
2A9A 3A4ØF4             LD      A,(BREAKLC)     ;<BREAK> hit ?
2A9D E6Ø4               AND     4
2A9F 28F8               JR      Z,RBLP          ;No - wait for interrupt
              ;
              ;         <BREAK> key hit - Abort
              ;
2AA1 F3                 DI                      ;Cancel next interrupt
2AA2 CDA92D             CALL    DISDOKI         ;Put *DO & *KI back
2AA5 CD2B2A             CALL    CASSOFF         ;Turn off cassette
2AA8 ØEØD               LD      C,CR            ;End line
2AAA CD4328             CALL    DSP
2AAD C35427             JP      ABORT           ;Go to abort routine
              ;
              ;         Interrupt Handler - Comes from RST 38
              ;
2ABØ C3B32A   RST38V    JP      $+3             ;Wait
2AB3 F5                 PUSH    AF              ;Save status
2AB4 DBEØ               IN      A,(PORTEØ)      ;Read port
2AB6 1F                 RRA                     ;Bit Ø low ?
```

```
2AB7 D2C12A     Ø554Ø           JP      NC,BITØLOW
2ABA 1F         Ø555Ø           RRA                     ;Bit 1 low ?
2ABB D2C52A     Ø556Ø           JP      NC,BIT1LOW
2ABE F1         Ø557Ø           POP     AF              ;Recover status
2ABF FB         Ø558Ø           EI                      ;Back on
2ACØ C9         Ø559Ø           RET                     ;RETurn
                Ø56ØØ ;
                Ø561Ø ;         Set E = bit image - bit Ø or 1
                Ø562Ø ;
2AC1 1EØ1       Ø563Ø BITØLOW   LD      E,1             ;High
2AC3 18Ø2       Ø564Ø           JR      BIT1LOW+2       ;Add interrupt offset
2AC5 1EØØ       Ø565Ø BIT1LOW   LD      E,Ø             ;Low
2AC7 3EØ6       Ø566Ø           LD      A,ROUTOFF       ;Add interrupt routine
2AC9 81         Ø567Ø           ADD     A,C             ;Offset to C
2ACA 4F         Ø568Ø           LD      C,A
                Ø569Ø ;
                Ø57ØØ ;         Is the Head on a valid pulse ?
                Ø571Ø ;
2ACB DBFF       Ø572Ø           IN      A,(PORTFF)      ;Read cassette level
2ACD E6Ø1       Ø573Ø           AND     1               ;Mask off all but bit Ø
2ACF BB         Ø574Ø           CP      E               ;Same as given level ?
2ADØ 2ØØ3       Ø575Ø           JR      NZ,WAITINT      ;No - wait for next inter
                Ø576Ø ;
                Ø577Ø ;         Valid pulse - Get out of interrupt routine
                Ø578Ø ;
2AD2 F1         Ø579Ø           POP     AF              ;Remove RST 38 RET addr
2AD3 F1         Ø58ØØ           POP     AF
2AD4 C9         Ø581Ø           RET
                Ø582Ø ;
                Ø583Ø ;         Not the right interrupt - wait for next
                Ø584Ø ;
2AD5 F1         Ø585Ø WAITINT   POP     AF              ;Recover status
2AD6 FB         Ø586Ø           EI                      ;  and wait for next
2AD7 C9         Ø587Ø           RET                     ;  interrupt
                Ø588Ø ;
                Ø589Ø ;         RDHEAD - Read a TAPE1ØØ header
                Ø59ØØ ;
2AD8 2A6627     Ø591Ø RDHEAD    LD      HL,(CURPOS)     ;P/u cursor position
2ADB 119727     Ø592Ø           LD      DE,BUFFER       ;Buffer
2ADE CD392B     Ø593Ø           CALL    RDSYNC          ;Read in SYNC
                Ø594Ø ;
                Ø595Ø ;         Read in Header Type byte
                Ø596Ø ;
2AE1 CDA72B     Ø597Ø           CALL    RDBYTE          ;Read type byte
2AE4 FE9C       Ø598Ø           CP      9CH             ;Text type ?
2AE6 2ØFØ       Ø599Ø           JR      NZ,RDHEAD       ;No - try again
                Ø6ØØØ ;
2AE8 Ø1ØØØ6     Ø6Ø1Ø           LD      BC,6ØØH         ;B=6 bytes, Checksum = Ø
                Ø6Ø2Ø ;
2AEB CDA22B     Ø6Ø3Ø RFNLP     CALL    RDBYTEC         ;Read byte
2AEE 77         Ø6Ø4Ø           LD      (HL),A          ;Save byte
2AEF 12         Ø6Ø5Ø           LD      (DE),A          ;Stuff in buffer
2AFØ 23         Ø6Ø6Ø           INC     HL              ;Bump cursor pos
2AF1 13         Ø6Ø7Ø           INC     DE              ;Bump buffer ptr
2AF2 1ØF7       Ø6Ø8Ø           DJNZ    RFNLP
                Ø6Ø9Ø ;
                Ø61ØØ ;         Next ten bytes are unused
                Ø611Ø ;
2AF4 Ø6ØA       Ø612Ø           LD      B,1Ø
2AF6 CDA22B     Ø613Ø BOGUSLP   CALL    RDBYTEC         ;Read byte & checksum
2AF9 1ØFB       Ø614Ø           DJNZ    BOGUSLP
```

```
                   ;
                   ;       Negate checksum
                   ;
2AFB 79                    LD      A,C             ;P/u checksum
2AFC ED44                  NEG                     ;Negate it
2AFE 4F                    LD      C,A
2AFF CDA72B                CALL    RDBYTE          ;Read in Checksum byte
2B02 B9                    CP      C               ;Match ?
2B03 C4232B                CALL    NZ,CHKERR       ;No - checksum error
                   ;
                   ;       Read in twenty zeros
                   ;
2B06 0614                  LD      B,20
2B08 CDA72B        DUMBYT  CALL    RDBYTE
2B0B 10FB                  DJNZ    DUMBYT
                   ;
                   ;       Check if this is the correct filename
                   ;
2B0D 00            CORRECT NOP                     ;X'C9' if first filename
2B0E 119727                LD      DE,BUFFER       ;Is this the one ?
2B11 219027                LD      HL,FILENM
2B14 0606                  LD      B,6             ;6 chars in filename
                   ;
                   ;       Loop to compare (HL) to (DE)
                   ;
2B16 1A            CKFILE  LD      A,(DE)          ;P/u header byte
2B17 CDB32C                CALL    CONV_UC         ;Convert to U/C
2B1A BE                    CP      (HL)            ;Match ?
2B1B 23                    INC     HL
2B1C 13                    INC     DE
2B1D C2D82A                JP      NZ,RDHEAD       ;No - try again
2B20 10F4                  DJNZ    CKFILE
2B22 C9                    RET                     ;Yes - RETurn
                   ;
                   ;       Checksum error - Either ignore it or "C"
                   ;
2B23 00            CHKERR  NOP                     ;RETurn or NOP
2B24 F3                    DI                      ;Disable interrupts
2B25 3E43                  LD      A,'C'           ;<C>hecksum error
2B27 324FF8        CHKERR2 LD      (VIDEO+79),A
2B2A CDA92D                CALL    DISDOKI         ;Bring back RAM
2B2D CD2B2A                CALL    CASSOFF         ;Turn off motor
2B30 216827                LD      HL,READERR      ;"Tape Read Error!"
2B33 CD4928                CALL    DSPLY
2B36 C35427                JP      ABORT           ;Good bye
                   ;
                   ;       RDSYNC - Read Cassette SYNC byte field
                   ;
                   ;       Save Registers
                   ;
2B39 E5            RDSYNC  PUSH    HL              ;Save regs
2B3A D5                    PUSH    DE
2B3B C5                    PUSH    BC
2B3C 3E01                  LD      A,1             ;Set interrupt vector
2B3E D3E0                  OUT     (PORTE0),A
                   ;
                   ;       Read in 128 bits (16 bytes) initially
                   ;
2B40 0680          RDSYNC2 LD      B,80H           ;Read 128 bits (16 bytes)
2B42 CD962A        RBTLP   CALL    RDBIT           ;Read bit
2B45 79                    LD      A,C             ;P/u count value
```

```
2B46 FEØF                CP      TOOSHRT         ;Is this a bit ?
2B48 38F6                JR      C,RDSYNC2       ;No - didn't find a bit
2B4A FE3E                CP      TOOLONG         ;Is this a bit ?
2B4C 3ØF2                JR      NC,RDSYNC2      ;No - wait for bit
2B4E 1ØF2                DJNZ    RBTLP           ;Legal bit - dec count
                ;
                ;        Now check parity of next 128 bits
                ;
2B5Ø 21ØØØØ     RESCNT   LD      HL,Ø            ;H = Ø's count, L = 1's
2B53 Ø64Ø                LD      B,4ØH
                ;
                ;        Read in 3 bits
                ;
2B55 CD962A     LOOP     CALL    RDBIT           ;Read bit
2B58 CD962A              CALL    RDBIT           ;Read bit
2B5B 51                  LD      D,C             ;Save count
2B5C CD962A              CALL    RDBIT           ;Read bit
                ;
                ;        Calculate Difference between last 2 bits
                ;
2B5F 7A                  LD      A,D             ;P/u last bit
2B6Ø 91                  SUB     C               ;Subtract current bit
2B61 3ØØ2                JR      NC,ABSVAL
2B63 ED44                NEG                     ;Change to ABS value
                ;
                ;        If Value < DIFFER then Bit = 1, else Bit = Ø
                ;
2B65 FEØD       ABSVAL   CP      DIFFER          ;Bit = 1 ?
2B67 38Ø3                JR      C,BIT1          ;Yes - bump Bit 1 count
2B69 24                  INC     H               ;No - bump Bit Ø count
2B6A 18Ø1                JR      DODJ            ;Back to loop
2B6C 2C         BIT1     INC     L               ;Bump Bit 1 count
2B6D 1ØE6       DODJ     DJNZ    LOOP            ;Dec count - go to loop
                ;
                ;        Check if H (Ø's count) & L (1's count) = 4Ø
                ;
2B6F 3E4Ø                LD      A,4ØH           ;Is H = 64 ?
2B71 BC                  CP      H
2B72 28ØA                JR      Z,CHKMARK       ;Yes - check for marker
2B74 BD                  CP      L               ;Is L = 64 ?
2B75 2ØD9                JR      NZ,RESCNT       ;No - Reset count
                ;
                ;        Set interrupt Vector & discard 1 bit
                ;
2B77 3EØ2                LD      A,2             ;Set interrupt vector
2B79 D3EØ                OUT     (PORTEØ),A
2B7B CD962A              CALL    RDBIT           ;Read bit
                ;
                ;        Rotate each bit read in D & check if = X'7F'
                ;
2B7E 16ØØ       CHKMARK  LD      D,Ø             ;Set byte = Ø
2B8Ø CD962A     GETBIT   CALL    RDBIT           ;Read next bit
2B83 CD8F2B              CALL    ROTBYTE         ;Rotate into Byte (D)
2B86 7A                  LD      A,D             ;P/u byte
2B87 FE7F                CP      7FH             ;Marker byte ?
2B89 2ØF5                JR      NZ,GETBIT       ;No - get another bit
                ;
                ;        Found marker byte - Restore Regs & RETurn
                ;
2B8B C1                  POP     BC              ;Restore Registers
2B8C D1                  POP     DE
```

```
2B8D E1         Ø737Ø          POP     HL
2B8E C9         Ø738Ø          RET                         ;Done
                Ø739Ø ;
                Ø74ØØ ;        ROTBYTE - Rotate bit through D & check if error
                Ø741Ø ;
2B8F 79         Ø742Ø ROTBYTE  LD      A,C                 ;P/u count
2B9Ø FE22       Ø743Ø          CP      WHICH1              ;Bit = Ø or 1 ?
2B92 CB12       Ø744Ø          RL      D                   ;Set bit if Carry set
2B94 FEØF       Ø745Ø          CP      TOOSHRT             ;Too quick ?
2B96 DA9C2B     Ø746Ø          JP      C,CIOERR            ;Yes - I/O Error
2B99 FE3E       Ø747Ø          CP      TOOLONG             ;Too long
2B9B D8         Ø748Ø          RET     C                   ;No - RETurn
                Ø749Ø ;
                Ø75ØØ ;        Cassette I/O Error - Display Error
                Ø751Ø ;
2B9C F3         Ø752Ø CIOERR   DI                          ;Interrupts off
2B9D 3E44       Ø753Ø          LD      A,'D'               ;Data Error
2B9F C3272B     Ø754Ø          JP      CHKERR2
                Ø755Ø ;
                Ø756Ø ;        RDBYTEC - Read byte & Add byte to Check Sum
                Ø757Ø ;
2BA2 CDA72B     Ø758Ø RDBYTEC  CALL    RDBYTE              ;Read byte
2BA5 81         Ø759Ø          ADD     A,C                 ;Add to checksum
2BA6 C9         Ø76ØØ          RET                         ;Done
                Ø761Ø ;
                Ø762Ø ;        RDBYTE - Read a byte
                Ø763Ø ;        A <= Byte
                Ø764Ø ;
2BA7 D5         Ø765Ø RDBYTE:  PUSH    DE                  ;Save regs
2BA8 C5         Ø766Ø          PUSH    BC
2BA9 CD962A     Ø767Ø          CALL    RDBIT               ;Get bogus bit
2BAC 16ØØ       Ø768Ø          LD      D,Ø                 ;Init byte = Ø
2BAE Ø6Ø8       Ø769Ø          LD      B,8                 ;8 bits to read
                Ø77ØØ ;
2BBØ CD962A     Ø771Ø RDBLP    CALL    RDBIT               ;Read a bit
2BB3 CD8F2B     Ø772Ø          CALL    ROTBYTE             ;Rotate into D
2BB6 1ØF8       Ø773Ø          DJNZ    RDBLP
                Ø774Ø ;
                Ø775Ø ;        Add to Byte count
                Ø776Ø ;
2BB8 3A5228     Ø777Ø          LD      A,(COUNT)           ;P/u count
2BBB 3C         Ø778Ø          INC     A                   ;  & inc it
2BBC E63F       Ø779Ø          AND     3FH                 ;Ck if the 64th
2BBE 325228     Ø78ØØ          LD      (COUNT),A           ;Save the count
2BC1 2ØØ8       Ø781Ø          JR      NZ,NOTBLNK
                Ø782Ø ;
2BC3 3A4FF8     Ø783Ø          LD      A,(VIDEO+79)        ;Blink every 64
2BC6 EEØA       Ø784Ø          XOR     ØAH
2BC8 324FF8     Ø785Ø          LD      (VIDEO+79),A
                Ø786Ø ;
2BCB 7A         Ø787Ø NOTBLNK  LD      A,D                 ;Xfer byte to A
2BCC 18ØØ       Ø788Ø          JR      NEXTINS             ;Timing
                Ø789Ø ;
2BCE C1         Ø79ØØ NEXTINS  POP     BC                  ;Restore BC & DE
2BCF D1         Ø791Ø          POP     DE
2BDØ C9         Ø792Ø          RET                         ;Done
                Ø793Ø ;
                Ø794Ø ;        WRBIT - Write a bit to Cassette
                Ø795Ø ;
                Ø796Ø ;        Set DE = Delay Count for bit
                Ø797Ø ;
```

```
2BD1 CBØ1     WRBIT   RLC     C                 ;Get bit
2BD3 3ØØ5             JR      NC,NOPULS         ;NC - bit Ø
2BD5 111712   BT1     LD      DE,DELAY1         ;Delay for bit 1
2BD8 18Ø3             JR      DEL_LP            ;Go to delay
2BDA 112F2B   NOPULS  LD      DE,DELAYØ         ;Delay for bit=Ø
              ;
              ;       Delay 18 counts for 1, 43 counts for Ø
              ;
2BDD 15       DEL_LP  DEC     D                 ;Dec count
2BDE 2ØFD             JR      NZ,DEL_LP
2BEØ 3EØ2             LD      A,2               ;Ø Volts to tape
2BE2 D3FF             OUT     (PORTFF),A
2BE4 1D       DEL_LP2 DEC     E                 ;Secondary delay
2BE5 2ØFD             JR      NZ,DEL_LP2
2BE7 3EØ1             LD      A,1               ;Ø.85 volts to tape
2BE9 D3FF             OUT     (PORTFF),A
2BEB C9               RET                       ;Done
              ;
              ;       WRHEAD - Write a cassette header
              ;
2BEC CD6Ø2C   WRHEAD  CALL    WRSYNC            ;Write SYNC pattern
              ;
              ;       Write Text header type byte X'9C'
              ;
2BEF 16ØØ             LD      D,Ø               ;Init checksum = Ø
2BF1 ØE9C             LD      C,9CH             ;Text header type byte
2BF3 CD512C           CALL    WRBYTE            ;Write type byte
              ;
              ;       Write Filename in header block
              ;
2BF6 Ø6Ø6             LD      B,6               ;B = 6 chars
2BF8 219Ø27           LD      HL,FILENM         ;HL => Filename
2BFB 4E       FILELP  LD      C,(HL)            ;P/u filename character
2BFC CD4A2C           CALL    WRBYTEC           ;  and write it
2BFF 23               INC     HL                ;Bump count
2CØØ 1ØF9             DJNZ    FILELP
              ;
              ;       Write 1Ø filler bytes
              ;
2CØ2 Ø6ØA             LD      B,1Ø
2CØ4 CD4A2C   BOGUS   CALL    WRBYTEC
2CØ7 1ØFB             DJNZ    BOGUS
              ;
              ;       Write checksum byte & 2Ø dummy X'ØØ' bytes
              ;
2CØ9 7A               LD      A,D               ;P/u checksum
2CØA ED44             NEG
2CØC 4F               LD      C,A               ;  & xfer to C
2CØD CD512C           CALL    WRBYTE            ;Write Checksum byte
2C1Ø Ø1ØØ14           LD      BC,14ØØH          ;B = 2Ø bytes, C = Ø
2C13 CD512C   DUMMY   CALL    WRBYTE            ;Write byte
2C16 1ØFB             DJNZ    DUMMY
2C18 C9               RET                       ;Get back quick
              ;
              ;       WRDAT - Write a chunk of data to cassette
              ;
2C19 21ØØ3Ø   WRDAT   LD      HL,MEM            ;HL => Mem start
2C1C CD272C   WRDAT2  CALL    WRDATA            ;Write Block
2C1F 24               INC     H
2C2Ø 3AF52D           LD      A,(FCB1+4)        ;Finished ?
2C23 BC               CP      H
```

```
2C24 2ØF6                     JR      NZ,WRDAT2         ;No - write another
2C26 C9                       RET                       ;Yes - RETurn
                   ;
                   ;          WRDATA - Write a data Block
                   ;          HL => 256 byte block of data (page boundary)
                   ;
2C27 CD6Ø2C        WRDATA     CALL    WRSYNC            ;Write sync pattern
2C2A ØE8D                     LD      C,8DH             ;Write X'8D' type byte
2C2C CD512C                   CALL    WRBYTE
                   ;
                   ;          Write 256 byte block of data
                   ;
2C2F AF                       XOR     A                 ;Set checksum = Ø
2C3Ø 4E            WBLP       LD      C,(HL)            ;P/u byte
2C31 81                       ADD     A,C               ;Add checksum
2C32 F5                       PUSH    AF                ;Save A
2C33 CD512C                   CALL    WRBYTE            ;Write byte
2C36 F1                       POP     AF                ;Recover checksum
2C37 2C                       INC     L                 ;Bump count
2C38 2ØF6                     JR      NZ,WBLP
                   ;
                   ;          Write checksum byte
                   ;
2C3A ED44                     NEG                       ;Negate checksum
2C3C 4F                       LD      C,A               ;Write checksum byte
2C3D CD512C                   CALL    WRBYTE
                   ;
                   ;          Write 2Ø dummy bytes - X'ØØ'
                   ;
2C4Ø Ø614                     LD      B,2Ø              ;Write 2Ø dummy zeroes
2C42 ØEØØ          WDLP       LD      C,Ø
2C44 CD512C                   CALL    WRBYTE
2C47 1ØF9                     DJNZ    WDLP
2C49 C9                       RET                       ;Done
                   ;
                   ;          WRBYTEC - Write a byte & add checksum
                   ;
2C4A CD512C        WRBYTEC    CALL    WRBYTE            ;Write byte
2C4D 79                       LD      A,C               ;P/u byte
2C4E 82                       ADD     A,D               ;Add checksum
2C4F 57                       LD      D,A               ;New checksum
2C5Ø C9                       RET                       ;And RETurn
                   ;
                   ;          WRBYTE - Write a byte to Cassette
                   ;          C => Byte to Output
                   ;
2C51 C5            WRBYTE:    PUSH    BC                ;Save regs
2C52 D5                       PUSH    DE
2C53 CDDA2B                   CALL    NOPULS            ;Write dummy pulse
2C56 Ø6Ø8                     LD      B,8               ;8 bits to write
2C58 CDD12B        WRBTLP     CALL    WRBIT             ;Write bit
2C5B 1ØFB                     DJNZ    WRBTLP
2C5D D1                       POP     DE                ;Restore regs
2C5E C1                       POP     BC
2C5F C9                       RET
                   ;
                   ;          WRSYNC - Write a SYNC pattern to Cassette
                   ;
2C6Ø F3            WRSYNC     DI                        ;Disable interrupts
2C61 C5                       PUSH    BC                ;Save BC
2C62 Ø68Ø                     LD      B,8ØH             ;Delay
```

```
2C64                 @@PAUSE
2C64 3E10                    LD      A,16
2C66 EF                      RST     40
2C67 015500                  LD      BC,0055H          ;B = 256, C = X'55'
                     ;
                     ;       Write SYNC bytes - X'55'
                     ;
2C6A CD762C          WR55LP  CALL    WRBYTE8           ;Write 8 bit byte
2C6D 10FB                    DJNZ    WR55LP
                     ;
                     ;       Write Marker byte - X'7F'
                     ;
2C6F 0E7F                    LD      C,7FH             ;Write marker byte X'7F'
2C71 CD762C                  CALL    WRBYTE8
2C74 C1                      POP     BC                ;Recover BC
2C75 C9                      RET                       ;Done
                     ;
2C76 C5              WRBYTE8 PUSH    BC                ;Save B
2C77 0608                    LD      B,8               ;8 bits long
2C79 CDD12B          WB8LP   CALL    WRBIT             ;Write bit
2C7C 10FB                    DJNZ    WB8LP
2C7E C1                      POP     BC
2C7F C9                      RET
2C80                 *GET    TAPE100B:3
                     ;TAPE100B/ASM - Disk I/O & other routines
                     ;
                     ;       DISPSTR - Display String
                     ;
2C80 D5              DISPSTR PUSH    DE                ;Save DE
2C81 ED5B6627                LD      DE,(CURPOS)       ;P/u cursor position
2C85 7E              DSLP    LD      A,(HL)            ;P/u source char
2C86 FE03                    CP      ETX               ;Done ?
2C88 2815                    JR      Z,EXIT1           ;Yes - exit
2C8A FE0D                    CP      CR                ;Done ?
2C8C 280E                    JR      Z,EXIT2           ;Yes - exit
2C8E FE0A                    CP      LF                ;Line feed ?
2C90 2005                    JR      NZ,STUFCHR        ;No - stuff character
2C92 CDA52C                  CALL    NEXTLIN           ;Get next line
2C95 1802                    JR      BUMPIT
2C97 12              STUFCHR LD      (DE),A            ;Output to video
2C98 13                      INC     DE
2C99 23              BUMPIT  INC     HL                ;No - bump count
2C9A 18E9                    JR      DSLP
2C9C CDA52C          EXIT2   CALL    NEXTLIN           ;Next one down
2C9F ED536627        EXIT1   LD      (CURPOS),DE       ;Save cursor position
2CA3 D1                      POP     DE                ;Restore DE
2CA4 C9                      RET
                     ;
                     ;       NEXTLIN - Position to next line on video
                     ;       DE => RAM location
                     ;
2CA5 E5              NEXTLIN PUSH    HL                ;Save regs
2CA6 EB                      EX      DE,HL             ;Xfer # to HL
2CA7 CDE32D                  CALL    GETCRS            ;Calculate X,Y
2CAA 24                      INC     H                 ;Bump row #
2CAB 2E00                    LD      L,0               ;  and start @ beginning
2CAD CDCF2D                  CALL    GETPOS2           ;Convert to RAM location
2CB0 EB                      EX      DE,HL             ;Stuff into DE
2CB1 E1                      POP     HL
2CB2 C9                      RET
                     ;
```

```
              Ø978Ø ;         CONV_UC - Convert A to upper case
              Ø979Ø ;
2CB3 FE61     Ø98ØØ CONV_UC CP       'a'              ;Lower case ?
2CB5 D8       Ø981Ø         RET      C                ;No
2CB6 FE7B     Ø982Ø         CP       'z'+1            ;Lower case ?
2CB8 DØ       Ø983Ø         RET      NC               ;No
2CB9 CBAF     Ø984Ø         RES      5,A              ;Convert to Upper Case
2CBB C9       Ø985Ø         RET
              Ø986Ø ;
              Ø987Ø ;         CURSOFF - Turn off Cursor
              Ø988Ø ;
2CBC F5       Ø989Ø CURSOFF PUSH     AF               ;Save regs
2CBD D5       Ø99ØØ         PUSH     DE
2CBE C5       Ø991Ø         PUSH     BC
2CBF ØEØF     Ø992Ø         LD       C,CUROFF         ;Cursor off Character
2CC1          Ø993Ø         @@DSP
2CC1 3EØ2     ØØØ47         LD       A,2
2CC3 EF       ØØØ48         RST      4Ø
2CC4 C24627   Ø994Ø         JP       NZ,IOERR
2CC7 C1       Ø995Ø         POP      BC               ;Restore regs
2CC8 D1       Ø996Ø         POP      DE
2CC9 F1       Ø997Ø         POP      AF
2CCA C9       Ø998Ø         RET
              Ø999Ø ;
              1ØØØØ ;         INIT - Init a file
              1ØØ1Ø ;
2CCB 3E3A     1ØØ2Ø INIT    LD       A,@INIT          ;SVC #
2CCD 18Ø6     1ØØ3Ø         JR       DOSVC            ;INIT file
              1ØØ4Ø ;
              1ØØ5Ø ;         OPEN - Open Source File
              1ØØ6Ø ;
2CCF FDCB12C6 1ØØ7Ø OPEN    SET      Ø,(IY+SFLAG$)    ;Inhibit file-open bit
2CD3 3E3B     1ØØ8Ø         LD       A,@OPEN          ;OPEN SVC #
              1ØØ9Ø ;
2CD5 F5       1Ø1ØØ DOSVC   PUSH     AF
2CD6 D5       1Ø11Ø         PUSH     DE
2CD7 219D27   1Ø12Ø         LD       HL,DFBUF         ;HL => Disk filename buf
2CDA 1A       1Ø13Ø TLP     LD       A,(DE)           ;P/u byte from FCB
2CDB 77       1Ø14Ø         LD       (HL),A           ;Xfer to TEMBUF
2CDC 23       1Ø15Ø         INC      HL
2CDD 13       1Ø16Ø         INC      DE
2CDE FEØE     1Ø17Ø         CP       CR+1             ;Done ?
2CEØ 38Ø8     1Ø18Ø         JR       C,DUN
2CE2 FE3A     1Ø19Ø         CP       ':'
2CE4 28Ø4     1Ø2ØØ         JR       Z,DUN
2CE6 FE2E     1Ø21Ø         CP       '.'
2CE8 2ØFØ     1Ø22Ø         JR       NZ,TLP
              1Ø23Ø ;
              1Ø24Ø ;         Found valid terminator - Is this a device ?
              1Ø25Ø ;
2CEA 2B       1Ø26Ø DUN     DEC      HL               ;Back up to term
2CEB D1       1Ø27Ø         POP      DE               ;DE => FCB+Ø
2CEC 1A       1Ø28Ø         LD       A,(DE)           ;Device ?
2CED FE2A     1Ø29Ø         CP       '*'
2CEF 28Ø7     1Ø3ØØ         JR       Z,DUN2           ;Yes - done
2CF1 363A     1Ø31Ø         LD       (HL),':'         ;No - overwrite with ":"
2CF3 23       1Ø32Ø         INC      HL               ;Bump
2CF4 22122D   1Ø33Ø         LD       (DSPEC+1),HL     ;Save drivespec location
2CF7 23       1Ø34Ø         INC      HL               ;Bump
2CF8 36Ø3     1Ø35Ø DUN2    LD       (HL),ETX         ;End with X'Ø3'
2CFA F1       1Ø36Ø         POP      AF               ;A = SVC #
```

```
2CFB 32212D               LD      (SVCNUM+1),A    ;Save SVC #
2CFE 21ØØ2F               LD      HL,IOBUFF       ;HL => I/O Buffer
2DØ1 Ø6ØØ                 LD      B,Ø             ;LRL = 256
2DØ3 EF                   RST     28H             ;OPEN or INIT file
2DØ4 28Ø3        CHECK    JR      Z,CHKPROT       ;Check PROTection status
                 ;
                 ;        Ignore Error #42 - "LRL Open Fault"
                 ;
2DØ6 FE2A                 CP      42              ;Ignore this error
2DØ8 CØ                   RET     NZ              ;NZ - Abort
                 ;
                 ;        Stuff Drive # into Buffer
                 ;
2DØ9 D5          CHKPROT  PUSH    DE              ;P/u drivespec
2DØA DDE1                 POP     IX              ;  from FCB+6
2DØC DD7EØ6               LD      A,(IX+6)
2DØF C63Ø                 ADD     A,'Ø'           ;Convert to ASCII
2D11 32ØØØØ      DSPEC    LD      ($-$),A
                 ;
                 ;        Check if File has proper Access
                 ;
2D14 DDCBØØ7E             BIT     7,(IX)          ;Is FCB open?
2D18 281F                 JR      Z,ILLFILE       ;No - Illegal Filename
2D1A DD7EØ1               LD      A,(IX+1)        ;P/u protection byte
2D1D E6Ø7                 AND     7
2D1F 47                   LD      B,A             ;Xfer to B
                 ;
2D2Ø 3EØØ        SVCNUM   LD      A,$-$           ;P/u SVC #
2D22 FE3A                 CP      @INIT           ;@INIT ?
2D24 78                   LD      A,B             ;P/u protection level
2D25 28ØC                 JR      Z,INIT1         ;Z - Must be < 5
2D27 FEØ6                 CP      6               ;Read Access ?
2D29 38ØC                 JR      C,OKYDOKY       ;Yes - set Z & RETurn
                 ;
                 ;        Illegal Access to protected file
                 ;
2D2B             ILLACC   @@CLOSE                 ;Close File
2D2B 3E3C                 LD      A,6Ø
2D2D EF                   RST     4Ø
2D2E 3E19                 LD      A,25            ;File Access Denied
2D3Ø C34627               JP      IOERR           ;Error - Regardless
                 ;
2D33 FEØ5        INIT1    CP      5               ;Update Access ?
2D35 3ØF4                 JR      NC,ILLACC       ;No - Illegal Access
2D37 AF          OKYDOKY  XOR     A               ;RETurn Z
2D38 C9                   RET
                 ;
2D39 3E13        ILLFILE  LD      A,19            ;Illegal Filename
2D3B B7                   OR      A               ;Set NZ
2D3C C9                   RET                     ;
                 ;
                 ;        CLOSE - Close the Destination File
                 ;
2D3D 11112E      CLOSE    LD      DE,FCB2         ;DE => FCB
2D4Ø                      @@CLOSE                 ;Close File
2D4Ø 3E3C                 LD      A,6Ø
2D42 EF                   RST     4Ø
2D43 C8                   RET     Z               ;Good - RETurn
2D44 C34627               JP      IOERR           ;Bad - Quit
                 ;
                 ;        WRITESC - Write a Sector to Destination file
```

```
             1Ø94Ø ;
2D47 11112E  1Ø95Ø WRITESC LD      DE,FCB2         ;DE => FCB
2D4A         1Ø96Ø         @@WRITE                 ;Write Sector
2D4A 3E4B    ØØØ53         LD      A,75
2D4C EF      ØØØ54         RST     4Ø
2D4D C24627  1Ø97Ø         JP      NZ,IOERR        ;Bad - quit
2D5Ø C9      1Ø98Ø         RET                     ;Good - RETurn
             1Ø99Ø ;
             11ØØØ ;       WRTDEST - Write Destination File
             11Ø1Ø ;
2D51 11112E  11Ø2Ø WRTDEST LD      DE,FCB2         ;DE => Destination FCB
2D54 21152E  11Ø3Ø WRTDES  LD      HL,FCB2+4       ;HL => msb of I/O buffer
2D57 34      11Ø4Ø         INC     (HL)            ;Bump
2D58 CD472D  11Ø5Ø         CALL    WRITESC         ;Write Sector
2D5B 3EØØ    11Ø6Ø EOTF2   LD      A,$-$           ;P/u # of sectors
2D5D BE      11Ø7Ø         CP      (HL)            ;Finished ?
2D5E 2ØF4    11Ø8Ø         JR      NZ,WRTDES       ;No - back to loop
             11Ø9Ø ;
             111ØØ ;       Finished Writing - Set EOF offset byte
             1111Ø ;
2D6Ø 3EØØ    1112Ø OFFSET  LD      A,$-$           ;P/u offset byte
2D62 32192E  1113Ø         LD      (FCB2+8),A      ;  & stuff into FCB
2D65 CD3D2D  1114Ø         CALL    CLOSE           ;Close the File
2D68 C9      1115Ø         RET
             1116Ø ;
             1117Ø ;       READSRC - Read in chunk of Source Disk file
             1118Ø ;
2D69 21F52D  1119Ø READSRC LD      HL,FCB1+4       ;HL => Hi byte of I/O buf
2D6C 362F    112ØØ         LD      (HL),MEM<-8-1   ;Init FCB I/O buffer
             1121Ø ;
             1122Ø ;       Read in Source file
             1123Ø ;
2D6E 11F12D  1124Ø READSR2 LD      DE,FCB1         ;Pt DE to FCB
2D71 34      1125Ø         INC     (HL)            ;Bump I/O buffer
2D72         1126Ø         @@READ                  ;Read a sector
2D72 3E43    ØØØ55         LD      A,67
2D74 EF      ØØØ56         RST     4Ø
2D75 28F7    1127Ø         JR      Z,READSR2
             1128Ø ;
             1129Ø ;       Fill remainder of sector w/ X'1A's
             113ØØ ;
2D77 F5      1131Ø         PUSH    AF              ;Save Error code
2D78 3AF92D  1132Ø NOMORE  LD      A,(FCB1+8)      ;P/u EOF offset byte
2D7B ED44    1133Ø         NEG
2D7D 47      1134Ø         LD      B,A             ;Xfer to B for DJNZ
2D7E 66      1135Ø         LD      H,(HL)          ;P/u I/O buffer msb
2D7F 2EFF    1136Ø         LD      L,ØFFH          ;End of sector
2D81 28Ø1    1137Ø         JR      Z,NULBUF        ;Z - keep HL here
2D83 25      1138Ø         DEC     H               ;Sector boundary
2D84 361A    1139Ø NULBUF  LD      (HL),1AH        ;Fill remainder of buffer
2D86 2B      114ØØ         DEC     HL              ;  with zeroes
2D87 1ØFB    1141Ø         DJNZ    NULBUF
             1142Ø ;
             1143Ø ;       Add a sector of 1As
             1144Ø ;
2D89 24      1145Ø         INC     H               ;Pt to next sector
2D8A 2EØØ    1146Ø         LD      L,Ø
2D8C 361A    1147Ø XTR1AS  LD      (HL),Ø1AH       ;EOF indicator
2D8E 23      1148Ø         INC     HL              ;Bump
2D8F 1ØFB    1149Ø         DJNZ    XTR1AS
2D91 F1      115ØØ DONTFIL POP     AF              ;Recover error code
```

```
               ;
               ;          I/O Error - Better be EOF error
               ;
2D92 FE1C                 CP      1CH              ;EOF ?
2D94 C8                   RET     Z                ;Yes - RETurn
2D95 FE1D                 CP      1DH              ;NRN > ERN
2D97 C8                   RET     Z                ;Yes - RETurn
2D98 C34627               JP      IOERR            ;No - Disk Error
               ;
               ;          ENDOKI - Enable Video & Keyboard
               ;
2D9B F5        ENDOKI     PUSH    AF
2D9C E5                   PUSH    HL
2D9D 3A78ØØ               LD      A,(OPREG$)       ;P/u port mask
2DAØ 32AC2D               LD      (SVOPREG+1),A    ;  and save it for DISDOKI
2DA3 CB87                 RES     Ø,A              ;Reset bit Ø
2DA5 CBCF                 SET     1,A              ;Set bit 1
2DA7 18Ø4                 JR      DOOPREG          ;Set new assignment
               ;
               ;          DISDOKI - Disable Video & Keyboard
               ;
2DA9 F5        DISDOKI    PUSH    AF
2DAA E5                   PUSH    HL
               ;
2DAB 3EØØ      SVOPREG    LD      A,$-$            ;Restore original mask
2DAD 3278ØØ    DOOPREG    LD      (OPREG$),A
2DBØ D384                 OUT     (@OPREG),A       ;  and disable video
               ;
2DB2 E1                   POP     HL               ;Restore regs & RETurn
2DB3 F1                   POP     AF
2DB4 C9                   RET
               ;
               ;          SWAP38 - Swap 38H - 3AH with save area
               ;
2DB5 Ø6Ø3      SWAP38     LD      B,3              ;3 bytes to exchange
2DB7 21C72D               LD      HL,SWAREA        ;HL => Swap Area
2DBA 1138ØØ               LD      DE,38H           ;DE => Restart Xfer addr
2DBD 4E        SWAPLP     LD      C,(HL)           ;P/u source
2DBE 1A                   LD      A,(DE)
2DBF EB                   EX      DE,HL            ;Swap ptrs
2DCØ 71                   LD      (HL),C           ;Stuff in dest
2DC1 12                   LD      (DE),A
2DC2 23                   INC     HL               ;Bump ptrs
2DC3 13                   INC     DE
2DC4 1ØF7                 DJNZ    SWAPLP           ;3 bytes to swap
2DC6 C9                   RET
               ;
2DC7 C3BØ2A    SWAREA     JP      RST38V           ;JP vector
               ;
               ;          GETPOS - Get current cursor position in video
               ;
2DCA Ø6Ø4      GETPOS     LD      B,4              ;P/u current cursor pos
2DCC                      @@VDCTL
2DCC 3EØF                 LD      A,15
2DCE EF                   RST     4Ø
2DCF 4D        GETPOS2    LD      C,L              ;Save column #
2DDØ 6C                   LD      L,H
2DD1 26ØØ                 LD      H,Ø              ;HL => Row #
2DD3 54                   LD      D,H              ;Set DE = HL
2DD4 5D                   LD      E,L
2DD5 29                   ADD     HL,HL            ;X 2
```

```
2DD6 29         121ØØ           ADD     HL,HL           ;X 4
2DD7 19         1211Ø           ADD     HL,DE           ;X 5
2DD8 29         1212Ø           ADD     HL,HL           ;X 1Ø
2DD9 29         1213Ø           ADD     HL,HL           ;X 2Ø
2DDA 29         1214Ø           ADD     HL,HL           ;X 4Ø
2DDB 29         1215Ø           ADD     HL,HL           ;X 8Ø
2DDC Ø6F8       1216Ø           LD      B,VIDEO<-8      ;D = high byte of video
2DDE Ø9         1217Ø           ADD     HL,BC           ;HL => Cursor location
2DDF 226627     1218Ø           LD      (CURPOS),HL     ;Save cursor position
2DE2 C9         1219Ø           RET
                122ØØ ;
                1221Ø ;         GETCRS - Calculate row x column cursor pos
                1222Ø ;         HL => Cursor position in RAM
                1223Ø ;         HL <= Cursor position in Row (H) Column (L)
                1224Ø ;
2DE3 11ØØF8     1225Ø GETCRS    LD      DE,VIDEO        ;Get offset
2DE6 B7         1226Ø           OR      A
2DE7 ED52       1227Ø           SBC     HL,DE
2DE9 ØE5Ø       1228Ø           LD      C,8Ø            ;Calculate row #
2DEB            1229Ø           @@DIV16
2DEB 3E5E       ØØØ59           LD      A,94
2DED EF         ØØØ6Ø           RST     4Ø
2DEE 65         123ØØ           LD      H,L             ;Set H = Row
2DEF 6F         1231Ø           LD      L,A             ;Set L = Column
2DFØ C9         1232Ø           RET
                Ø43ØØ ;
ØØ2Ø            Ø431Ø FCB1      DS      32
ØØ2Ø            Ø432Ø FCB2      DS      32
ØØ19            Ø433Ø INBUFF    DS      25
                Ø434Ø ;
2FØØ            Ø435Ø           ORG     $<-8+1<+8
                Ø436Ø ;
Ø1ØØ            Ø437Ø IOBUFF    DS      256
3ØØØ            Ø438Ø MEM       EQU     $
                Ø439Ø ;
26ØØ            Ø44ØØ           END     START
```

```
@@1            ØØØØ @@2            ØØØØ @@3            ØØØØ
@@4            ØØØØ @INIT          ØØ3A @MOD2          ØØØØ
@MOD4          FFFF @OPEN          ØØ3B @OPREG         ØØ84
ABB            ØØ1Ø ABORT          2754 ABSVAL         2B65
AFTER          2A73 AP             ØØ27 BITØLOW        2AC1
BIT1           2B6C BIT1LOW        2AC5 BOGUS          2CØ4
BOGUSLP        2AF6 BREAK          ØØ8Ø BREAKLC        F44Ø
BS             ØØØ8 BT1            2BD5 BUFFER         2797
BUMPIT         2C99 CASSOFF        2A2B CASSON         2A1A
CFLAG$         ØØØ2 CHECK          2DØ4 CHKERR         2B23
CHKERR2        2B27 CHKMARK        2B7E CHKPRM         261A
CHKPROT        2DØ9 CHKSEC         26C6 CIOERR         2B9C
CKFILE         2B16 CKPLP          27E9 CLEAN          27B2
CLOSE          2D3D CONV_UC        2CB3 CORRECT        2BØD
COUNT          2852 CPARM          27FD CR             ØØØD
CRESP          29EØ CUROFF         ØØØF CURON          ØØØE
CURPOS         2766 CURSOFF        2CBC DDF            2941
DELAYØ         2B2F DELAY1         1217 DEL_LP         2BDD
DEL_LP2        2BE4 DFBUF          279D DFLAG$         ØØØ3
DIFFER         ØØØD DISDOKI        2DA9 DISPSTR        2C8Ø
DLEN           2763 DODJ           2B6D DOINIT         27C7
DOINPUT        2816 DONTFIL        2D91 DOOPREG        2DAD
DOSVC          2CD5 DSF            29ØA DSLP           2C85
DSP            2843 DSPEC          2D11 DSPLY          2849
DUMBYT         2BØ8 DUMMY          2C13 DUN            2CEA
DUN2           2CF8 DUNLIN         27FC ENDOKI         2D9B
ENUF           266Ø EOTF           2A51 EOTF2          2D5B
ETX            ØØØ3 EXDSP          284D EXIT           2758
EXIT1          2C9F EXIT2          2C9C FCB1           2DF1
FCB2           2E11 FILELP         2BFB FILENM         279Ø
FLAG           ØØ4Ø FORNOW         273A GETBIT         2B8Ø
GETCRS         2DE3 GETFILN        27BA GETPOS         2DCA
GETPOS2        2DCF GTFILE         27AC GTFILE2        26E5
HELLO$         2853 ILLACC         2D2B ILLEGAL        2751
ILLFILE        2D39 INBUFF         2E31 INIT           2CCB
INIT1          2D33 INPUT          2835 INP_R_W        2621
IOBUFF         2FØØ IOERR          2746 KFLAG$         ØØØA
LF             ØØØA LOADA          ØØ3A LOOP           2B55
MEM            3ØØØ MODMASK        ØØØC MODOUT         ØØEC
NEXTINS        2BCE NEXTLIN        2CA5 NOMORE         2D78
NOPULS         2BDA NOTBLNK        2BCB NOTENT         2A41
NULBUF         2D84 NUM            ØØ8Ø OFFSET         2D6Ø
OKYDOKY        2D37 OLDSP          275B OPDSRC         264F
OPEN           2CCF OPREG$         ØØ78 PARMTBL        29C8
PAR_ERR        ØØ2C PORTEØ         ØØEØ PORTFF         ØØFF
PRDEST         281Ø PRDEST2        282D PRMERR         29BB
PRMERR$        297F PRSOUR         28Ø8 PRSOUR2        2825
PRTAPE         2A3B RBLP           2A99 RBTLP          2B42
RDBIT          2A96 RDBLP          2BBØ RDBYTE         2BA7
RDBYTEC        2BA2 RDDAT          2A49 RDDAT2         2A4C
RDDATA         2A58 RDHEAD         2AD8 RDLP1          2A66
RDLP2          2A8D RDORWR         28DB RDSYNC         2B39
RDSYNC2        2B4Ø RDTAPE         26B1 READERR        2768
READFIL        26EB READING        277B READSR2        2D6E
READSRC        2D69 RESCNT         2B5Ø RFNLP          2AEB
ROTBYTE        2B8F ROUTOFF        ØØØ6 RPARM          29E4
RRESP          29D7 RST38V         2ABØ SFLAG$         ØØ12
SKPSPC         26D4 START          26ØØ STARTA         26Ø9
STR            ØØ2Ø STUFCHR        2C97 SVCNUM         2D2Ø
SVOPREG        2DAB SWAP38         2DB5 SWAPLP         2DBD
```

```
SWAREA      2DC7 TAB        ØØØ9 TDF        2923
TLP         2CDA TOOBIG     29BF TOOBIG$    299Ø
TOOLONG     ØØ3E TOOSHRT    ØØØF TREADY     295F
TSF         28F1 USEHI      27D6 VFLAG$     ØØ15
VIDEO       F8ØØ WAITINT    2AD5 WB8LP      2C79
WBLP        2C3Ø WDLP       2C42 WHICH1     ØØ22
WPARM       29E6 WR55LP     2C6A WRBIT      2BD1
WRBTLP      2C58 WRBYTE     2C51 WRBYTE8    2C76
WRBYTEC     2C4A WRDAT      2C19 WRDAT2     2C1C
WRDATA      2C27 WRESP      29CF WRHEAD     2BEC
WRITESC     2D47 WRITING    2785 WRMASK     ØØ16
WRSYNC      2C6Ø WRTAPE     263A WRTAPE2    2643
WRTDES      2D54 WRTDES2    274Ø WRTDEST    2D51
XTR1AS      2D8C @@ABORT    926F @@ADTSK    93Ø2
@@BANK      981A @@BKSP     94FA @@BREAK    983Ø
@@CHNIO     925A @@CKBRKC   987E @@CKDRV    9356
@@CKEOF     95ØF @@CKTSK    92ED @@CLOSE    94E5
@@CLS       9868 @@CMNDI    9299 @@CMNDR    92AE
@@CTL       9ØBE @@DATE     923Ø @@DCSTAT   9395
@@DEBUG     92D8 @@DECHEX   979A @@DIRRD    97Ø7
@@DIRWR     971C @@DIV16    9785 @@DIV8     977Ø
@@DODIR     936B @@DSP      9Ø82 @@DSPLY    9122
@@ERROR     92C3 @@EXIT     9284 @@FEXT     9674
@@FLAGS     98Ø4 @@FNAME    9689 @@FSPEC    965F
@@GATRD     96F2 @@GATWR    9731 @@GET      9Ø96
@@GTDCB     96B3 @@GTDCT    969E @@GTMOD    96C8
@@HDFMT     943D @@HEX16    97D9 @@HEX8     97C4
@@HEXDEC    97AF @@HIGH$    97EE @@INIT     94BB
@@KBD       9ØFA @@KEY      9Ø6E @@KEYIN    91ØE
@@KLTSK     9341 @@LOAD     9635 @@LOC      9524
@@LOF       9539 @@LOGER    9159 @@LOGOT    916E
@@MSG       91A5 @@MUL16    975B @@MUL8     9746
@@OPEN      94DØ @@PARAM    921B @@PAUSE    92Ø6
@@PEOF      954E @@POSN     9563 @@PRINT    91BA
@@PRT       9ØD2 @@PUT      9ØAA @@RAMDIR   938Ø
@@RDSEC     9413 @@RDSSC    96DD @@READ     9578
@@REMOV     94A6 @@RENAM    9491 @@REW      958D
@@RMTSK     9317 @@RPTSK    932C @@RREAD    95A2
@@RSLCT     93FE @@RSTOR    93BF @@RUN      964A
@@RWRIT     95B7 @@SEEK     93E9 @@SEEKSC   95CC
@@SKIP      95E1 @@SLCT     93AA @@STEPI    93D4
@@TIME      9245 @@VDCTL    91F1 @@VER      95F6
@@VRSEC     9428 @@WEOF     96ØB @@WHERE    9ØE6
@@WRITE     962Ø @@WRSEC    9452 @@WRSSC    9467
@@WRTRK     947C
26ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

NOTES:

NOTES:

NOTES:

NOTES:

NOTES:

```
                00100 ;LDOS6Ø/EQU - Equates from cross reference of Lowcore
ØØØØ            ØØ11Ø         TITLE   <LDOS6Ø/EQU>
                ØØ12Ø ;
Ø8FØ            ØØ13Ø @$SYS   EQU     Ø8FØH
ØØØØ            ØØ14Ø @@1     DEFL    ØØØØH
ØØØØ            ØØ15Ø @@2     DEFL    ØØØØH
ØØØØ            ØØ16Ø @@3     DEFL    ØØØØH
ØØØØ            ØØ17Ø @@4     DEFL    ØØØØH
Ø877            ØØ18Ø @BANK   EQU     Ø877H
13ØØ            ØØ19Ø @BYTEIO EQU     13ØØH
Ø689            ØØ2ØØ @CHNIO  EQU     Ø689H
Ø553            ØØ21Ø @CKBRKC EQU     Ø553H
Ø545            ØØ22Ø @CLS    EQU     Ø545H
Ø623            ØØ23Ø @CTL    EQU     Ø623H
Ø7A8            ØØ24Ø @DATE   EQU     Ø7A8H
Ø6E3            ØØ25Ø @DIV16  EQU     Ø6E3H
Ø642            ØØ26Ø @DSP    EQU     Ø642H
Ø52D            ØØ27Ø @DSPLY  EQU     Ø52DH
ØØØØ            ØØ28Ø @FRENCH EQU     ØØØØH
ØØØØ            ØØ29Ø @GERMAN EQU     ØØØØH
Ø638            ØØ3ØØ @GET    EQU     Ø638H
Ø7BD            ØØ31Ø @HEX16  EQU     Ø7BDH
Ø7C2            ØØ32Ø @HEX8   EQU     Ø7C2H
Ø6F6            ØØ33Ø @HEXDEC EQU     Ø6F6H
ØØØØ            ØØ34Ø @HZ5Ø   EQU     ØØØØH
ØØØØ            ØØ35Ø @INTL   EQU     ØØØØH
Ø63Ø            ØØ36Ø @JCL    EQU     Ø63ØH
Ø635            ØØ37Ø @KBD    EQU     Ø635H
Ø628            ØØ38Ø @KEY    EQU     Ø628H
Ø585            ØØ39Ø @KEYIN  EQU     Ø585H
ØØ89            ØØ4ØØ @KITSK  EQU     ØØ89H
Ø5Ø3            ØØ41Ø @LOGER  EQU     Ø5Ø3H
Ø5ØØ            ØØ42Ø @LOGOT  EQU     Ø5ØØH
ØØØØ            ØØ43Ø @MOD2   EQU     ØØØØH
FFFF            ØØ44Ø @MOD4   EQU     ØFFFFH
Ø53Ø            ØØ45Ø @MSG    EQU     Ø53ØH
Ø6C9            ØØ46Ø @MUL16  EQU     Ø6C9H
ØØ84            ØØ47Ø @OPREG  EQU     ØØ84H
Ø528            ØØ48Ø @PRINT  EQU     Ø528H
Ø63D            ØØ49Ø @PRT    EQU     Ø63DH
Ø645            ØØ5ØØ @PUT    EQU     Ø645H
ØFE9            ØØ51Ø @RSTNMI EQU     ØFE9H
Ø68Ø            ØØ52Ø @RSTREG EQU     Ø68ØH
Ø78D            ØØ53Ø @TIME   EQU     Ø78DH
FFFF            ØØ54Ø @USA    EQU     ØFFFFH
ØB99            ØØ55Ø @VDCTL  EQU     ØB99H
ØD38            ØØ56Ø @VDCTL3 EQU     ØD38H
ØD42            ØØ57Ø @_VDCTL EQU     ØD42H
ØDF1            ØØ58Ø ADDR_2_ROWCOL   EQU     ØDF1H
Ø2Ø1            ØØ59Ø BAR$    EQU     Ø2Ø1H
439D            ØØ6ØØ BOOTST$ EQU     439DH
Ø2ØØ            ØØ61Ø BUR$    EQU     Ø2ØØH
ØA7B            ØØ62Ø CASHK$  EQU     ØA7BH
ØØ6C            ØØ63Ø CFLAG$  EQU     ØØ6CH
Ø3ØØ            ØØ64Ø CORE$   DEFL    Ø3ØØH
F8ØØ            ØØ65Ø CRTBGN$ EQU     ØF8ØØH
ØØ33            ØØ66Ø DATE$   EQU     ØØ33H
Ø4C7            ØØ67Ø DAYTBL$ EQU     Ø4C7H
ØØ31            ØØ68Ø DCBKL$  EQU     ØØ31H
Ø47Ø            ØØ69Ø DCT$    EQU     Ø47ØH
ØØ6D            ØØ7ØØ DFLAG$  EQU     ØØ6DH
```

```
0846          00710 DIS_DO_RAM      EQU     0846H
0B94          00720 DODATA$ EQU     0B94H
0210          00730 DODCB$  EQU     0210H
0C44          00740 DO_CONTROL      EQU     0C44H
0CB8          00750 DO_DSPCHAR      EQU     0CB8H
0C8C          00760 DO_INVERT_DIS   EQU     0C8CH
0C89          00770 DO_INVERT_ENA   EQU     0C89H
0C9B          00780 DO_INVERT_OFF   EQU     0C9BH
0000          00790 DO_MASK EQU     0000H
0BCB          00800 DO_RET  EQU     0BCBH
0BCC          00810 DO_RET1 EQU     0BCCH
0CCE          00820 DO_SCROLL       EQU     0CCEH
0BEA          00830 DO_TABS EQU     0BEAH
04C0          00840 DSKTYP$ EQU     04C0H
04C2          00850 DTPMT$  EQU     04C2H
0FF4          00860 DVREND$ EQU     0FF4H
0206          00870 DVRHI$  EQU     0206H
0817          00880 ENADIS_DO_RAM   EQU     0817H
000E          00890 FDDINT$ EQU     000EH
006A          00900 FLGTAB$ EQU     006AH
0DAE          00910 GET_@_ROWCOL    EQU     0DAEH
0750          00920 HERTZ$  EQU     0750H
040E          00930 HIGH$   EQU     040EH
0072          00940 IFLAG$  EQU     0072H
0420          00950 INBUF$  EQU     0420H
003E          00960 INTVC$  EQU     003EH
0203          00970 JCLCB$  EQU     0203H
0230          00980 JLDCB$  EQU     0230H
07D6          00990 KCK@    EQU     07D6H
0074          01000 KFLAG$  EQU     0074H
08FC          01010 KIDATA$ EQU     08FCH
0208          01020 KIDCB$  EQU     0208H
0202          01030 LBANK$  EQU     0202H
0401          01040 MAXDAY$ EQU     0401H
0076          01050 MODOUT$ EQU     0076H
04DC          01060 MONTBL$ EQU     04DCH
0077          01070 NFLAG$  EQU     0077H
0078          01080 OPREG$  EQU     0078H
086E          01090 OPREG_SV_AREA   EQU     086EH
0835          01100 OPREG_SV_PTR    EQU     0835H
0410          01110 PAKNAM$ EQU     0410H
0382          01120 PAUSE@  EQU     0382H
07AF          01130 PCSAVE$ EQU     07AFH
001B          01140 PDRV$   EQU     001BH
0218          01150 PRDCB$  EQU     0218H
0DCD          01160 PUTA@DE EQU     0DCDH
0DCA          01170 PUT_@   EQU     0DCAH
0DC6          01180 PUT_@_ROWCOL    EQU     0DC6H
007B          01190 RFLAG$  EQU     007BH
0DD0          01200 ROWCOL_2_ADDR   EQU     0DD0H
04C4          01210 RSTOR$  EQU     04C4H
0238          01220 S1DCB$  EQU     0238H
0CF3          01230 SET_SCROLL      EQU     0CF3H
007C          01240 SFLAG$  EQU     007CH
0220          01250 SIDCB$  EQU     0220H
0228          01260 SODCB$  EQU     0228H
0380          01270 STACK$  EQU     0380H
0000          01280 START$  EQU     0000H
002D          01290 TIME$   EQU     002DH
002C          01300 TIMER$  EQU     002CH
002B          01310 TIMSL$  EQU     002BH
```

```
Ø713           Ø132Ø TIMTSK$ EQU     Ø713H
Ø4C3           Ø133Ø TMPMT$  EQU     Ø4C3H
Ø7B1           Ø134Ø TRACE_INT       EQU      Ø7B1H
ØA8F           Ø135Ø TYPHK$  EQU     ØA8FH
ØB26           Ø136Ø TYPTSK$ EQU     ØB26H
ØØ7F           Ø137Ø VFLAG$  EQU     ØØ7FH
Ø4Ø1           Ø138Ø ZERO$   EQU     Ø4Ø1H
No end statement
ØØØØØ Total errors
```

```
           ;SYSØ/EQU - Equates from cross reference of Sysres
ØØØØ                TITLE   <SYSØ/EQU>
           ;
Ø3B7       $A1      EQU     Ø3B7H
Ø3B8       $A2      EQU     Ø3B8H
Ø3B9       $A3      EQU     Ø3B9H
147Ø       $CKEOF   EQU     147ØH
Ø8FØ       @$SYS    EQU     Ø8FØH
ØØØØ       @@1      DEFL    ØØØØH
ØØØØ       @@1      DEFL    ØØØØH
ØØØØ       @@2      DEFL    ØØØØH
ØØØØ       @@2      DEFL    ØØØØH
ØØØØ       @@3      DEFL    ØØØØH
ØØØØ       @@3      DEFL    ØØØØH
ØØØØ       @@4      DEFL    ØØØØH
ØØØØ       @@4      DEFL    ØØØØH
1BØ8       @ABORT   EQU     1BØ8H
1CDA       @ADTSK   EQU     1CDAH
Ø877       @BANK    EQU     Ø877H
1486       @BKSP    EQU     1486H
196F       @BREAK   EQU     196FH
13ØØ       @BYTEIO  EQU     13ØØH
Ø689       @CHNIO   EQU     Ø689H
Ø553       @CKBRKC  EQU     Ø553H
1993       @CKDRV   EQU     1993H
158F       @CKEOF   EQU     158FH
1CF5       @CKTSK   EQU     1CF5H
1999       @CLOSE   EQU     1999H
Ø545       @CLS     EQU     Ø545H
197E       @CMNDI   EQU     197EH
197B       @CMNDR   EQU     197BH
Ø623       @CTL     EQU     Ø623H
Ø7A8       @DATE    EQU     Ø7A8H
199F       @DBGHK   EQU     199FH
19CØ       @DCINIT  EQU     19CØH
19C4       @DCRES   EQU     19C4H
19B5       @DCSTAT  EQU     19B5H
1A2B       @DCTBYT  EQU     1A2BH
19AØ       @DEBUG   EQU     19AØH
Ø3E1       @DECHEX  EQU     Ø3E1H
18F7       @DIRCYL  EQU     18F7H
18BB       @DIRRD   EQU     18BBH
18Ø3       @DIRWR   EQU     18Ø3H
Ø6E3       @DIV16   EQU     Ø6E3H
1927       @DIV8    EQU     1927H
19AF       @DODIR   EQU     19AFH
19A9       @DOKEY   EQU     19A9H
Ø642       @DSP     EQU     Ø642H
Ø52D       @DSPLY   EQU     Ø52DH
1BØF       @ERROR   EQU     1BØFH
1BØB       @EXIT    EQU     1BØBH
1984       @FEXT    EQU     1984H
196A       @FLAGS   EQU     196AH
199C       @FNAME   EQU     199CH
ØØØØ       @FRENCH  EQU     ØØØØH
1981       @FSPEC   EQU     1981H
1874       @GATRD   EQU     1874H
1875       @GATWR   EQU     1875H
ØØØØ       @GERMAN  EQU     ØØØØH
Ø638       @GET     EQU     Ø638H
199Ø       @GTDCB   EQU     199ØH
```

```
1A1E        ØØ71Ø @GTDCT  EQU      1A1EH
19B2        ØØ72Ø @GTMOD  EQU      19B2H
19E4        ØØ73Ø @HDFMT  EQU      19E4H
Ø7BD        ØØ74Ø @HEX16  EQU      Ø7BDH
Ø7C2        ØØ75Ø @HEX8   EQU      Ø7C2H
Ø6F6        ØØ76Ø @HEXDEC EQU      Ø6F6H
1948        ØØ77Ø @HIGH$  EQU      1948H
1897        ØØ78Ø @HITRD  EQU      1897H
1898        ØØ79Ø @HITWR  EQU      1898H
ØØØØ        ØØ8ØØ @HZ5Ø   EQU      ØØØØH
ØØ86        ØØ81Ø @ICNFG  EQU      ØØ86H
198D        ØØ82Ø @INIT   EQU      198DH
ØØØØ        ØØ83Ø @INTL   EQU      ØØØØH
1BF2        ØØ84Ø @IPL    EQU      1BF2H
Ø63Ø        ØØ85Ø @JCL    EQU      Ø63ØH
Ø635        ØØ86Ø @KBD    EQU      Ø635H
Ø628        ØØ87Ø @KEY    EQU      Ø628H
Ø585        ØØ88Ø @KEYIN  EQU      Ø585H
ØØ89        ØØ89Ø @KITSK  EQU      ØØ89H
ØØ89        ØØ9ØØ @KITSK  EQU      ØØ89H
1CDØ        ØØ91Ø @KLTSK  EQU      1CDØH
1B38        ØØ92Ø @LOAD   EQU      1B38H
14B3        ØØ93Ø @LOC    EQU      14B3H
14DE        ØØ94Ø @LOF    EQU      14DEH
Ø5Ø3        ØØ95Ø @LOGER  EQU      Ø5Ø3H
Ø5ØØ        ØØ96Ø @LOGOT  EQU      Ø5ØØH
ØØØØ        ØØ97Ø @MOD2   EQU      ØØØØH
FFFF        ØØ98Ø @MOD4   EQU      ØFFFFH
Ø53Ø        ØØ99Ø @MSG    EQU      Ø53ØH
Ø6C9        Ø1ØØØ @MUL16  EQU      Ø6C9H
19ØA        Ø1Ø1Ø @MUL8   EQU      19ØAH
ØØ66        Ø1Ø2Ø @NMI    EQU      ØØ66H
198A        Ø1Ø3Ø @OPEN   EQU      198AH
ØØ84        Ø1Ø4Ø @OPREG  EQU      ØØ84H
1987        Ø1Ø5Ø @PARAM  EQU      1987H
Ø382        Ø1Ø6Ø @PAUSE  EQU      Ø382H
14A2        Ø1Ø7Ø @PEOF   EQU      14A2H
1434        Ø1Ø8Ø @POSN   EQU      1434H
Ø528        Ø1Ø9Ø @PRINT  EQU      Ø528H
Ø63D        Ø11ØØ @PRT    EQU      Ø63DH
Ø645        Ø111Ø @PUT    EQU      Ø645H
19AC        Ø112Ø @RAMDIR EQU      19ACH
19D8        Ø113Ø @RDHDR  EQU      19D8H
19F4        Ø114Ø @RDSEC  EQU      19F4H
18D8        Ø115Ø @RDSSC  EQU      18D8H
19EØ        Ø116Ø @RDTRK  EQU      19EØH
1513        Ø117Ø @READ   EQU      1513H
19A6        Ø118Ø @REMOVE EQU      19A6H
1996        Ø119Ø @RENAME EQU      1996H
149B        Ø12ØØ @REW    EQU      149BH
1CD7        Ø121Ø @RMTSK  EQU      1CD7H
1CEB        Ø122Ø @RPTSK  EQU      1CEBH
1473        Ø123Ø @RREAD  EQU      1473H
19D4        Ø124Ø @RSLCT  EQU      19D4H
ØØØØ        Ø125Ø @RSTØØ  EQU      ØØØØH
ØØØ8        Ø126Ø @RSTØ8  EQU      ØØØ8H
ØØ1Ø        Ø127Ø @RST1Ø  EQU      ØØ1ØH
ØØ18        Ø128Ø @RST18  EQU      ØØ18H
ØØ2Ø        Ø129Ø @RST2Ø  EQU      ØØ2ØH
ØØ28        Ø13ØØ @RST28  EQU      ØØ28H
ØØ3Ø        Ø131Ø @RST3Ø  EQU      ØØ3ØH
```

```
ØØ38        @RST38  EQU     ØØ38H
ØFE9        @RSTNMI EQU     ØFE9H
19C8        @RSTOR  EQU     19C8H
Ø68Ø        @RSTREG EQU     Ø68ØH
1B1D        @RUN    EQU     1B1DH
13AD        @RWRIT  EQU     13ADH
19DØ        @SEEK   EQU     19DØH
1421        @SEEKSC EQU     1421H
143Ø        @SKIP   EQU     143ØH
19BC        @SLCT   EQU     19BCH
Ø392        @SOUND  EQU     Ø392H
19CC        @STEPI  EQU     19CCH
Ø78D        @TIME   EQU     Ø78DH
FFFF        @USA    EQU     ØFFFFH
ØB99        @VDCTL  EQU     ØB99H
ØD38        @VDCTL3 EQU     ØD38H
156Ø        @VER    EQU     156ØH
19DC        @VRSEC  EQU     19DCH
14EC        @WEOF   EQU     14ECH
1979        @WHERE  EQU     1979H
1531        @WRITE  EQU     1531H
19E8        @WRSEC  EQU     19E8H
19EC        @WRSSC  EQU     19ECH
19FØ        @WRTRK  EQU     19FØH
ØD42        @_VDCTL EQU     ØD42H
ØDF1        ADDR_2_ROWCOL   EQU     ØDF1H
ØØ6A        AFLAG$  EQU     ØØ6AH
1FF1        AUTO?   EQU     1FF1H
Ø2Ø1        BAR$    EQU     Ø2Ø1H
439D        BOOTST$ EQU     439DH
1C6Ø        BREAK?  EQU     1C6ØH
1C88        BRKVEC$ EQU     1C88H
Ø2ØØ        BUR$    EQU     Ø2ØØH
ØA7B        CASHK$  EQU     ØA7BH
ØØEØ        CFCB$   EQU     ØØEØH
ØØEØ        CFGFCB$ EQU     ØØEØH
ØØ6C        CFLAG$  EQU     ØØ6CH
ØØ6C        CFLAG$  EQU     ØØ6CH
1A7F        CKMOD@  EQU     1A7FH
1568        CKOPEN@ EQU     1568H
2Ø3F        CONFIG$ EQU     2Ø3FH
1CFF        CORE$   DEFL    1CFFH
1BFF        CORE$   DEFL    1BFFH
1948        CORE$   DEFL    1948H
1948        CORE$   DEFL    1948H
Ø3ØØ        CORE$   DEFL    Ø3ØØH
F8ØØ        CRTBGN$ EQU     ØF8ØØH
16AE        CYL_GRN EQU     16AEH
1A26        D@FBYT8 EQU     1A26H
ØØ33        DATE$   EQU     ØØ33H
ØØ33        DATE$   EQU     ØØ33H
Ø4C7        DAYTBL$ EQU     Ø4C7H
ØØAØ        DBGSV$  EQU     ØØAØH
ØØ31        DCBKL$  EQU     ØØ31H
Ø47Ø        DCT$    EQU     Ø47ØH
1A29        DCTBYT8@        EQU     1A29H
1A34        DCTFLD@ EQU     1A34H
ØØ6D        DFLAG$  EQU     ØØ6DH
ØØ6D        DFLAG$  EQU     ØØ6DH
23ØØ        DIRBUF$ EQU     23ØØH
Ø846        DIS_DO_RAM      EQU     Ø846H
```

```
ØB94          Ø193Ø DODATA$ EQU     ØB94H
Ø21Ø          Ø194Ø DODCB$  EQU     Ø21ØH
ØC44          Ø195Ø DO_CONTROL      EQU     ØC44H
ØCB8          Ø196Ø DO_DSPCHAR      EQU     ØCB8H
ØC8C          Ø197Ø DO_INVERT_DIS   EQU     ØC8CH
ØC89          Ø198Ø DO_INVERT_ENA   EQU     ØC89H
ØC9B          Ø199Ø DO_INVERT_OFF   EQU     ØC9BH
ØØØØ          Ø2ØØØ DO_MASK EQU     ØØØØH
ØBCB          Ø2Ø1Ø DO_RET  EQU     ØBCBH
ØBCC          Ø2Ø2Ø DO_RET1 EQU     ØBCCH
ØCCE          Ø2Ø3Ø DO_SCROLL       EQU     ØCCEH
ØBEA          Ø2Ø4Ø DO_TABS EQU     ØBEAH
Ø4CØ          Ø2Ø5Ø DSKTYP$ EQU     Ø4CØH
Ø4C2          Ø2Ø6Ø DTPMT$  EQU     Ø4C2H
ØFF4          Ø2Ø7Ø DVREND$ EQU     ØFF4H
Ø2Ø6          Ø2Ø8Ø DVRHI$  EQU     Ø2Ø6H
ØØ6E          Ø2Ø9Ø EFLAG$  EQU     ØØ6EH
Ø817          Ø21ØØ ENADIS_DO_RAM   EQU     Ø817H
19A4          Ø211Ø EXTDBG$ EQU     19A4H
ØØØE          Ø212Ø FDDINT$ EQU     ØØØEH
ØØØE          Ø213Ø FDDINT$ EQU     ØØØEH
ØØ6F          Ø214Ø FEMSK$  EQU     ØØ6FH
ØØ6A          Ø215Ø FLGTAB$ EQU     ØØ6AH
ØØ6A          Ø216Ø FLGTAB$ EQU     ØØ6AH
ØDAE          Ø217Ø GET_@_ROWCOL    EQU     ØDAEH
Ø75Ø          Ø218Ø HERTZ$  EQU     Ø75ØH
Ø4ØE          Ø219Ø HIGH$   EQU     Ø4ØEH
1A6C          Ø22ØØ HKRES$  EQU     1A6CH
ØØ72          Ø221Ø IFLAG$  EQU     ØØ72H
ØØ72          Ø222Ø IFLAG$  EQU     ØØ72H
Ø42Ø          Ø223Ø INBUF$  EQU     Ø42ØH
ØØ3C          Ø224Ø INTIM$  EQU     ØØ3CH
ØØ3D          Ø225Ø INTMSK$ EQU     ØØ3DH
ØØ3E          Ø226Ø INTVC$  EQU     ØØ3EH
ØØ3E          Ø227Ø INTVC$  EQU     ØØ3EH
Ø2Ø3          Ø228Ø JCLCB$  EQU     Ø2Ø3H
ØØ24          Ø229Ø JDCB$   EQU     ØØ24H
ØØCØ          Ø23ØØ JFCB$   EQU     ØØCØH
Ø23Ø          Ø231Ø JLDCB$  EQU     Ø23ØH
ØØ26          Ø232Ø JRET$   EQU     ØØ26H
Ø7D6          Ø233Ø KCK@    EQU     Ø7D6H
ØØ74          Ø234Ø KFLAG$  EQU     ØØ74H
ØØ74          Ø235Ø KFLAG$  EQU     ØØ74H
Ø8FC          Ø236Ø KIDATA$ EQU     Ø8FCH
Ø2Ø8          Ø237Ø KIDCB$  EQU     Ø2Ø8H
Ø2Ø2          Ø238Ø LBANK$  EQU     Ø2Ø2H
ØØ23          Ø239Ø LDRV$   EQU     ØØ23H
ØØ75          Ø24ØØ LFLAG$  EQU     ØØ75H
1566          Ø241Ø LNKFCB@ EQU     1566H
ØØ1E          Ø242Ø LOW$    EQU     ØØ1EH
ØØØD          Ø243Ø LSVC$   EQU     ØØØDH
24ØØ          Ø244Ø MAXCOR$ EQU     24ØØH
Ø4Ø1          Ø245Ø MAXDAY$ EQU     Ø4Ø1H
3ØØØ          Ø246Ø MINCOR$ EQU     3ØØØH
ØØ76          Ø247Ø MODOUT$ EQU     ØØ76H
ØØ76          Ø248Ø MODOUT$ EQU     ØØ76H
Ø4DC          Ø249Ø MONTBL$ EQU     Ø4DCH
ØØ77          Ø25ØØ NFLAG$  EQU     ØØ77H
ØØ78          Ø251Ø OPREG$  EQU     ØØ78H
ØØ78          Ø252Ø OPREG$  EQU     ØØ78H
Ø86E          Ø253Ø OPREG_SV_AREA   EQU     Ø86EH
```

```
Ø835          Ø254Ø OPREG_SV_PTR    EQU     Ø835H
14DC          Ø255Ø ORARET@ EQU     14DCH
ØØ3B          Ø256Ø OSRLS$  EQU     ØØ3BH
ØØ85          Ø257Ø OSVER$  EQU     ØØ85H
ØØ69          Ø258Ø OVRLY$  EQU     ØØ69H
Ø41Ø          Ø259Ø PAKNAM$ EQU     Ø41ØH
Ø382          Ø26ØØ PAUSE@  EQU     Ø382H
Ø7AF          Ø261Ø PCSAVE$ EQU     Ø7AFH
ØØ1B          Ø262Ø PDRV$   EQU     ØØ1BH
ØØ1B          Ø263Ø PDRV$   EQU     ØØ1BH
ØØ1C          Ø264Ø PHIGH$  EQU     ØØ1CH
Ø218          Ø265Ø PRDCB$  EQU     Ø218H
ØDCD          Ø266Ø PUTA@DE EQU     ØDCDH
ØDCA          Ø267Ø PUT_@   EQU     ØDCAH
ØDC6          Ø268Ø PUT_@_ROWCOL    EQU     ØDC6H
ØØ7B          Ø269Ø RFLAG$  EQU     ØØ7BH
ØØ7B          Ø27ØØ RFLAG$  EQU     ØØ7BH
ØDDØ          Ø271Ø ROWCOL_2_ADDR   EQU     ØDDØH
1BFF          Ø272Ø RST38@  EQU     1BFFH
Ø4C4          Ø273Ø RSTOR$  EQU     Ø4C4H
13A2          Ø274Ø RWRIT@  EQU     13A2H
Ø238          Ø275Ø S1DCB$  EQU     Ø238H
1DØØ          Ø276Ø SBUFF$  EQU     1DØØH
1A79          Ø277Ø SET@EXEC        EQU     1A79H
ØCF3          Ø278Ø SET_SCROLL      EQU     ØCF3H
ØØ8C          Ø279Ø SFCB$   EQU     ØØ8CH
ØØ7C          Ø28ØØ SFLAG$  EQU     ØØ7CH
ØØ7C          Ø281Ø SFLAG$  EQU     ØØ7CH
Ø22Ø          Ø282Ø SIDCB$  EQU     Ø22ØH
Ø228          Ø283Ø SODCB$  EQU     Ø228H
2142          Ø284Ø SPACE4$ EQU     2142H
Ø38Ø          Ø285Ø STACK$  EQU     Ø38ØH
ØØØØ          Ø286Ø START$  EQU     ØØØØH
ØØØØ          Ø287Ø START$  EQU     ØØØØH
ØØØB          Ø288Ø SVCRET$ EQU     ØØØBH
Ø1ØØ          Ø289Ø SVCTAB$ EQU     Ø1ØØH
1B13          Ø29ØØ SYSERR$ EQU     1B13H
ØØ4E          Ø291Ø TCB$    EQU     ØØ4EH
ØØ7D          Ø292Ø TFLAG$  EQU     ØØ7DH
ØØ2D          Ø293Ø TIME$   EQU     ØØ2DH
ØØ2D          Ø294Ø TIME$   EQU     ØØ2DH
ØØ2C          Ø295Ø TIMER$  EQU     ØØ2CH
ØØ2C          Ø296Ø TIMER$  EQU     ØØ2CH
ØØ2B          Ø297Ø TIMSL$  EQU     ØØ2BH
ØØ2B          Ø298Ø TIMSL$  EQU     ØØ2BH
Ø713          Ø299Ø TIMTSK$ EQU     Ø713H
Ø4C3          Ø3ØØØ TMPMT$  EQU     Ø4C3H
Ø7B1          Ø3Ø1Ø TRACE_INT       EQU     Ø7B1H
ØA8F          Ø3Ø2Ø TYPHK$  EQU     ØA8FH
ØB26          Ø3Ø3Ø TYPTSK$ EQU     ØB26H
ØØ13          Ø3Ø4Ø USTOR$  EQU     ØØ13H
ØØ7F          Ø3Ø5Ø VFLAG$  EQU     ØØ7FH
ØØ7F          Ø3Ø6Ø VFLAG$  EQU     ØØ7FH
ØØ8Ø          Ø3Ø7Ø WRINT$  EQU     ØØ8ØH
Ø4Ø1          Ø3Ø8Ø ZERO$   EQU     Ø4Ø1H
13AØ          Ø3Ø9Ø ZEROA@  EQU     13AØH
No end statement
ØØØØØ Total errors
```

```
          ;SVCMAC/ASM - LS-DOS Version VI
0000                TITLE   <SVCMAC - MACRO EQUIVALENTS>
          ;*LIST    OFF
          ;
0000      @MOD2     EQU     0
FFFF      @MOD4     EQU     -1
0000      @@KEY     MACRO
0000                LD      A,1
0000                RST     40
0000                ENDM
0000      @@DSP     MACRO
0000                LD      A,2
0000                RST     40
0000                ENDM
0000      @@GET     MACRO
0000                LD      A,3
0000                RST     40
0000                ENDM
0000      @@PUT     MACRO
0000                LD      A,4
0000                RST     40
0000                ENDM
0000      @@CTL     MACRO
0000                LD      A,5
0000                RST     40
0000                ENDM
0000      @@PRT     MACRO
0000                LD      A,6
0000                RST     40
0000                ENDM
0000      @@WHERE   MACRO
0000                LD      A,7
0000                RST     40
0000                ENDM
0000      @@KBD     MACRO
0000                LD      A,8
0000                RST     40
0000                ENDM
0000      @@KEYIN   MACRO
0000                LD      A,9
0000                RST     40
0000                ENDM
0000      @@DSPLY   MACRO   #MSG
0000                IFEQ    %%,1
0000                LD      HL,#MSG
0000                ENDIF
0000                LD      A,10
0000                RST     40
0000                ENDM
0000      @@LOGER   MACRO
0000                LD      A,11
0000                RST     40
0000                ENDM
0000      @@LOGOT   MACRO   #MSG
0000                IFEQ    %%,1
0000                LD      HL,#MSG
0000                ENDIF
0000                LD      A,12
0000                RST     40
0000                ENDM
0000      @@MSG     MACRO
```

```
0000            00710          LD      A,13
0000            00720          RST     40
0000            00730          ENDM
0000            00740 @@PRINT  MACRO   #MSG
0000            00750          IFEQ    %%,1
0000            00760          LD      HL,#MSG
0000            00770          ENDIF
0000            00780          LD      A,14
0000            00790          RST     40
0000            00800          ENDM
0000            00810 @@VDCTL  MACRO
0000            00820          LD      A,15
0000            00830          RST     40
0000            00840          ENDM
0000            00850 @@PAUSE  MACRO
0000            00860          LD      A,16
0000            00870          RST     40
0000            00880          ENDM
0000            00890 @@PARAM  MACRO
0000            00900          LD      A,17
0000            00910          RST     40
0000            00920          ENDM
0000            00930 @@DATE   MACRO
0000            00940          LD      A,18
0000            00950          RST     40
0000            00960          ENDM
0000            00970 @@TIME   MACRO
0000            00980          LD      A,19
0000            00990          RST     40
0000            01000          ENDM
0000            01010 @@CHNIO  MACRO
0000            01020          LD      A,20
0000            01030          RST     40
0000            01040          ENDM
0000            01050 @@ABORT  MACRO
0000            01060          LD      A,21
0000            01070          RST     40
0000            01080          ENDM
0000            01090 @@EXIT   MACRO
0000            01100          LD      A,22
0000            01110          RST     40
0000            01120          ENDM
0000            01130 @@CMNDI  MACRO
0000            01140          LD      A,24
0000            01150          RST     40
0000            01160          ENDM
0000            01170 @@CMNDR  MACRO
0000            01180          LD      A,25
0000            01190          RST     40
0000            01200          ENDM
0000            01210 @@ERROR  MACRO
0000            01220          LD      A,26
0000            01230          RST     40
0000            01240          ENDM
0000            01250 @@DEBUG  MACRO
0000            01260          LD      A,27
0000            01270          RST     40
0000            01280          ENDM
0000            01290 @@CKTSK  MACRO
0000            01300          LD      A,28
0000            01310          RST     40
```

```
0000          01320         ENDM
0000          01330 @@ADTSK MACRO
0000          01340         LD      A,29
0000          01350         RST     40
0000          01360         ENDM
0000          01370 @@RMTSK MACRO
0000          01380         LD      A,30
0000          01390         RST     40
0000          01400         ENDM
0000          01410 @@RPTSK MACRO
0000          01420         LD      A,31
0000          01430         RST     40
0000          01440         ENDM
0000          01450 @@KLTSK MACRO
0000          01460         LD      A,32
0000          01470         RST     40
0000          01480         ENDM
0000          01490 @@CKDRV MACRO
0000          01500         LD      A,33
0000          01510         RST     40
0000          01520         ENDM
0000          01530 @@DODIR MACRO
0000          01540         LD      A,34
0000          01550         RST     40
0000          01560         ENDM
0000          01570 @@RAMDIR        MACRO
0000          01580         LD      A,35
0000          01590         RST     40
0000          01600         ENDM
0000          01610 @@DCSTAT        MACRO
0000          01620         LD      A,40
0000          01630         RST     40
0000          01640         ENDM
0000          01650 @@SLCT  MACRO
0000          01660         LD      A,41
0000          01670         RST     40
0000          01680         ENDM
0000          01690 @@RSTOR MACRO
0000          01700         LD      A,44
0000          01710         RST     40
0000          01720         ENDM
0000          01730 @@STEPI MACRO
0000          01740         LD      A,45
0000          01750         RST     40
0000          01760         ENDM
0000          01770 @@SEEK  MACRO
0000          01780         LD      A,46
0000          01790         RST     40
0000          01800         ENDM
0000          01810 @@RSLCT MACRO
0000          01820         LD      A,47
0000          01830         RST     40
0000          01840         ENDM
0000          01850 @@RDSEC MACRO
0000          01860         LD      A,49
0000          01870         RST     40
0000          01880         ENDM
0000          01890 @@VRSEC MACRO
0000          01900         LD      A,50
0000          01910         RST     40
0000          01920         ENDM
```

```
ØØØØ          @@HDFMT MACRO
ØØØØ                  LD      A,52
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@WRSEC MACRO
ØØØØ                  LD      A,53
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@WRSSC MACRO
ØØØØ                  LD      A,54
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@WRTRK MACRO
ØØØØ                  LD      A,55
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@RENAM MACRO
ØØØØ                  LD      A,56
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@REMOV MACRO
ØØØØ                  LD      A,57
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@INIT  MACRO
ØØØØ                  LD      A,58
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@OPEN  MACRO
ØØØØ                  LD      A,59
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@CLOSE MACRO
ØØØØ                  LD      A,6Ø
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@BKSP  MACRO
ØØØØ                  LD      A,61
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@CKEOF MACRO
ØØØØ                  LD      A,62
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@LOC   MACRO
ØØØØ                  LD      A,63
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@LOF   MACRO
ØØØØ                  LD      A,64
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@PEOF  MACRO
ØØØØ                  LD      A,65
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@POSN  MACRO
ØØØØ                  LD      A,66
ØØØØ                  RST     4Ø
ØØØØ                  ENDM
ØØØØ          @@READ  MACRO
```

```
0000                     LD      A,67
0000                     RST     40
0000                     ENDM
0000           @@REW     MACRO
0000                     LD      A,68
0000                     RST     40
0000                     ENDM
0000           @@RREAD   MACRO
0000                     LD      A,69
0000                     RST     40
0000                     ENDM
0000           @@RWRIT   MACRO
0000                     LD      A,70
0000                     RST     40
0000                     ENDM
0000           @@SEEKSC          MACRO
0000                     LD      A,71
0000                     RST     40
0000                     ENDM
0000           @@SKIP    MACRO
0000                     LD      A,72
0000                     RST     40
0000                     ENDM
0000           @@VER     MACRO
0000                     LD      A,73
0000                     RST     40
0000                     ENDM
0000           @@WEOF    MACRO
0000                     LD      A,74
0000                     RST     40
0000                     ENDM
0000           @@WRITE   MACRO
0000                     LD      A,75
0000                     RST     40
0000                     ENDM
0000           @@LOAD    MACRO
0000                     LD      A,76
0000                     RST     40
0000                     ENDM
0000           @@RUN     MACRO
0000                     LD      A,77
0000                     RST     40
0000                     ENDM
0000           @@FSPEC   MACRO
0000                     LD      A,78
0000                     RST     40
0000                     ENDM
0000           @@FEXT    MACRO
0000                     LD      A,79
0000                     RST     40
0000                     ENDM
0000           @@FNAME   MACRO
0000                     LD      A,80
0000                     RST     40
0000                     ENDM
0000           @@GTDCT   MACRO
0000                     LD      A,81
0000                     RST     40
0000                     ENDM
0000           @@GTDCB   MACRO
0000                     LD      A,82
```

```
ØØØØ          Ø315Ø         RST     4Ø
ØØØØ          Ø316Ø         ENDM
ØØØØ          Ø317Ø @@GTMOD MACRO
ØØØØ          Ø318Ø         LD      A,83
ØØØØ          Ø319Ø         RST     4Ø
ØØØØ          Ø32ØØ         ENDM
ØØØØ          Ø321Ø @@RDSSC MACRO
ØØØØ          Ø322Ø         LD      A,85
ØØØØ          Ø323Ø         RST     4Ø
ØØØØ          Ø324Ø         ENDM
ØØØØ          Ø325Ø @@GATRD MACRO
ØØØØ          Ø326Ø         LD      A,86
ØØØØ          Ø327Ø         RST     4Ø
ØØØØ          Ø328Ø         ENDM
ØØØØ          Ø329Ø @@DIRRD MACRO
ØØØØ          Ø33ØØ         LD      A,87
ØØØØ          Ø331Ø         RST     4Ø
ØØØØ          Ø332Ø         ENDM
ØØØØ          Ø333Ø @@DIRWR MACRO
ØØØØ          Ø334Ø         LD      A,88
ØØØØ          Ø335Ø         RST     4Ø
ØØØØ          Ø336Ø         ENDM
ØØØØ          Ø337Ø @@GATWR MACRO
ØØØØ          Ø338Ø         LD      A,89
ØØØØ          Ø339Ø         RST     4Ø
ØØØØ          Ø34ØØ         ENDM
ØØØØ          Ø341Ø @@MUL8  MACRO
ØØØØ          Ø342Ø         LD      A,9Ø
ØØØØ          Ø343Ø         RST     4Ø
ØØØØ          Ø344Ø         ENDM
ØØØØ          Ø345Ø @@MUL16 MACRO
ØØØØ          Ø346Ø         LD      A,91
ØØØØ          Ø347Ø         RST     4Ø
ØØØØ          Ø348Ø         ENDM
ØØØØ          Ø349Ø @@DIV8  MACRO
ØØØØ          Ø35ØØ         LD      A,93
ØØØØ          Ø351Ø         RST     4Ø
ØØØØ          Ø352Ø         ENDM
ØØØØ          Ø353Ø @@DIV16 MACRO
ØØØØ          Ø354Ø         LD      A,94
ØØØØ          Ø355Ø         RST     4Ø
ØØØØ          Ø356Ø         ENDM
ØØØØ          Ø357Ø @@DECHEX        MACRO
ØØØØ          Ø358Ø         LD      A,96
ØØØØ          Ø359Ø         RST     4Ø
ØØØØ          Ø36ØØ         ENDM
ØØØØ          Ø361Ø @@HEXDEC        MACRO
ØØØØ          Ø362Ø         LD      A,97
ØØØØ          Ø363Ø         RST     4Ø
ØØØØ          Ø364Ø         ENDM
ØØØØ          Ø365Ø @@HEX8  MACRO
ØØØØ          Ø366Ø         LD      A,98
ØØØØ          Ø367Ø         RST     4Ø
ØØØØ          Ø368Ø         ENDM
ØØØØ          Ø369Ø @@HEX16 MACRO
ØØØØ          Ø37ØØ         LD      A,99
ØØØØ          Ø371Ø         RST     4Ø
ØØØØ          Ø372Ø         ENDM
ØØØØ          Ø373Ø @@HIGH$ MACRO
ØØØØ          Ø374Ø         LD      A,1ØØ
ØØØØ          Ø375Ø         RST     4Ø
```

```
ØØØØ          Ø376Ø          ENDM
ØØØØ          Ø377Ø @@FLAGS  MACRO
ØØØØ          Ø378Ø          LD       A,1Ø1
ØØØØ          Ø379Ø          RST      4Ø
ØØØØ          Ø38ØØ          ENDM
ØØØØ          Ø381Ø @@BANK   MACRO
ØØØØ          Ø382Ø          LD       A,1Ø2
ØØØØ          Ø383Ø          RST      4Ø
ØØØØ          Ø384Ø          ENDM
ØØØØ          Ø385Ø @@BREAK  MACRO    #ADR
ØØØØ          Ø386Ø          IFEQ     %%,1
ØØØØ          Ø387Ø          LD       HL,#ADR
ØØØØ          Ø388Ø          ENDIF
ØØØØ          Ø389Ø          LD       A,1Ø3
ØØØØ          Ø39ØØ          RST      4Ø
ØØØØ          Ø391Ø          ENDM
ØØØØ          Ø392Ø @@CLS    MACRO
ØØØØ          Ø393Ø          LD       A,1Ø5
ØØØØ          Ø394Ø          RST      4Ø
ØØØØ          Ø395Ø          ENDM
ØØØØ          Ø396Ø @@CKBRKC          MACRO
ØØØØ          Ø397Ø          LD       A,1Ø6
ØØØØ          Ø398Ø          RST      4Ø
ØØØØ          Ø399Ø          ENDM
              Ø4ØØØ *LIST    ON
ØØØØ          Ø4Ø1Ø          END
ØØØØØ Total errors
```